

Министерство просвещения Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Уральский государственный педагогический университет»  
Институт математики, физики, информатики и технологий  
Кафедра информатики, информационных технологий  
и методики обучения информатике

# **ИНФОРМАЦИОННАЯ СИСТЕМА ПО УПРАВЛЕНИЮ МАТЕРИАЛЬНОЙ БАЗОЙ ОРГАНИЗАЦИИ**

*Выпускная квалификационная работа  
бакалавра по направлению подготовки  
09.03.02 – Информационные системы и технологии*

Исполнитель: студент группы ИСиТ-1601  
ИМФИиТ  
Шилков А.А.

Руководитель: к.п.н., доцент кафедры  
ИИТиМОИ  
Кудрявцев А.В.

Работа допущена к защите  
«\_\_\_\_» мая 2020 г.  
Зав. кафедрой \_\_\_\_\_

Екатеринбург – 2020

## Реферат

Шилков А.А. ИНФОРМАЦИОННАЯ СИСТЕМА ПО УПРАВЛЕНИЮ МАТЕРИАЛЬНОЙ БАЗОЙ ПРЕДПРИЯТИЯ, выпускная квалификационная работа: 52 стр., рис. 44, табл. 1, библи. 28 назв., приложений 0.

*Ключевые слова:* ИНФОРМАЦИОННАЯ СИСТЕМА, АВТОМАТИЗАЦИЯ, ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ, РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ, БАЗА ДАННЫХ.

*Объект разработки* – информационная система управления материальной базой предприятия.

*Цель работы* – разработка информационной системы, обеспечивающей автоматизацию управления материальной базой предприятия.

В работе описаны результаты проектирования и реализации информационной системы, которая обеспечивает управление материальной базой предприятия, хранение объектов материальной базы в сетевой базе данных, а также web-интерфейса для взаимодействия с системой с разграничением прав доступа пользователей.

## Оглавление

<b>ВВЕДЕНИЕ .....</b>	<b>4</b>
<b>ГЛАВА 1. АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ ОРГАНИЗАЦИИ .....</b>	<b>6</b>
1.1    ПОСТАНОВКА ЗАДАЧИ.....	6
1.2    ИНСТРУМЕНТЫ И МЕТОДЫ РАЗРАБОТКИ ИНФОРМАЦИОННЫХ СИСТЕМ.....	8
1.3    ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ.....	17
<b>ГЛАВА 2. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ПО УПРАВЛЕНИЮ МАТЕРИАЛЬНОЙ БАЗОЙ ПРЕДПРИЯТИЯ .....</b>	<b>21</b>
2.1    МОДЕЛЬНЫЕ ПРЕДСТАВЛЕНИЯ РАЗРАБОТАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ	21
2.2    ОПИСАНИЕ РАБОТЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПО УПРАВЛЕНИЮ МАТЕРИАЛЬНОЙ БАЗОЙ ПРЕДПРИЯТИЯ .....	36
2.3    РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ.....	40
2.3.1    Начало работы.....	40
2.3.2    Окно авторизации. ....	40
2.3.3    Рабочее окно пользователя.....	42
2.3.4    Рабочее окно администратора. ....	47
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>50</b>
<b>СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....</b>	<b>51</b>

## **Введение**

В данное время информационные системы управления, известные так же как ИСУ или MIS (Management Information System), являются основным инструментом для обработки различной информации, позволяя накапливать и хранить определенные знания в единых базах данных, а также предоставлять быстрый доступ к ним. Таким образом, становится очевидной важность разработки собственной информационной системы для управления материальной базой предприятия. Поскольку любое предприятие имеет какую-либо материальную базу, то оно так или иначе сталкивается с необходимостью вести учет и управление ресурсами, которое не всегда возможно вести вручную при помощи бумажного документооборота. Именно тут на помощь приходит информационная система управления, которая позволяет заменить часть бумажного документооборота при учете и контроле ресурсов предприятия, кроме того, ИСУ позволяет мгновенно получить любую необходимую информацию.

**Актуальность разработки** заключается в необходимости предприятий вести управление своей материальной базой, наиболее эффективный способ выполнения данной цели – разработка и использование собственной информационной системы управления.

**Цель разработки** – разработать информационную систему для управления материальной базой предприятия.

### **Задачи:**

1. Произвести терминологический анализ, а также выяснить состояние проблемы и провести анализ подходов к её решению.
2. Произвести анализ и обосновать выбор технологий реализации и необходимых программных платформ.
3. В соответствии с техническим заданием провести разработку системы на основе выбранных технологий.

4. Подготовить техническую и сопроводительную документацию.

# **Глава 1. Анализ информационных систем организации**

## **1.1 Постановка задачи**

Для начала необходимо выяснить что такое материальная база предприятия, чтобы в дальнейшем понимать с чем именно мы должны работать.

Материальная база предприятия — это комплекс технологического, энергетического, транспортного и других видов оборудования, инструментов и приспособлений, зданий и сооружений, необходимых для осуществления процесса производства. [24]

Любое предприятие имеет в своём распоряжении множество различного оборудования от обычных столов и стульев до компьютеров, различных станков, машин и даже зданий, которые могут находиться в нескольких филиалах предприятия, в том числе в разных городах или даже странах. Всё это приводит к тому, что возникает необходимость управлять данными ресурсами, как минимум вести их учет и следить за состоянием, кроме того рано или поздно может появиться необходимость расширения предприятия или же обновления материальных ресурсов из-за их поломок, устаревания или появления новых задач, для которых, например, имеющиеся станки могут не подходить по тем или иным характеристикам и/или функционалу.

Именно по этой причине возникает потребность в создании функциональной и простой информационной системы, позволяющей осуществлять управление материальной базой предприятия.

Что такое информационная система? В статье 2 Федерального закона от 27.07.2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации» сказано следующее: «информационная система — совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств.» [28]

Однако, нас более интересует что же такое информационная система управления. Информационная система управления (ИСУ) является совокупностью организационных, технических, программных и информационных средств, которые объединены в единую систему с целью сбора, хранения, обработки и выдачи информации, которая предназначена для выполнения функций управления. [23]

Таким образом, информационная система по управлению материальной базой предприятия будет представлять из себя ничто иное как некую базу данных или возможно даже несколько баз данных, если такая потребность возникнет, некое приложение, которое будет обеспечивать обработку информации, хранящуюся в данной базе или базах, а также некое техническое средство на котором данное приложение будет работать.

Поскольку основным источником информации для базы данных будут являться сотрудники, система должна включать в себя функции заполнения, редактирования и удаления данных пользователями. Данные функции могут быть реализованы на основе одной из реляционных систем управления базами данных, или СУБД.

СУБД – специализированная программа (чаще комплекс программ), предназначенная для организации и ведения базы данных. На сегодняшний день самыми популярными реляционными СУБД являются: PostgreSQL, MySQL, MongoDB, SAP HANA, Oracle Database и Microsoft SQL Server. [22]

Пытаясь найти аналоги каких-либо информационных систем подобной направленности, было выяснено, что самостоятельные готовые продукты в свободном доступе отсутствуют, однако, подобный функционал можно встретить в различных комплексных приложениях, но чаще всего они предоставляются по достаточно высокой цене и могут иметь более широкий функционал, чем необходимо.

Так, например, приложение «1С: Торговля и склад» в самой дешевой комплектации стоит 15 тысяч рублей за комплект, требует обучения сотрудников работе с данным приложением и обладает не самой простой настройкой для действительного упрощения работы.

На основании приведенных выше фактов можно сделать вывод, что разработка собственного продукта оправдана потребностью предприятий в собственной удобной, простой и функциональной системе для управления собственной материальной базой.

## **1.2 Инструменты и методы разработки информационных систем**

Поскольку основой любой информационной системы является база данных, то необходимо выбрать СУБД для дальнейшей работы с базами данных. Как уже было сказано, самыми распространенными СУБД на данный момент являются:

- PostgreSQL;
- MySQL;
- MongoDB;
- SAP HANA;
- Oracle;
- Microsoft SQL Server.

Однако, часть из них поставляется исключительно на платной основе, к ним относятся **SAP HANA, Oracle** и **Microsoft SQL Server**. Поскольку перед нами стоит задача создать информационную систему по управлению любым предприятием, то для небольших или очень ограниченных в бюджете предприятий затраты на поддержание данных СУБД могут оказаться гораздо выше получаемой выгоды, поэтому мы сразу исключим данные варианты из списка и рассмотрим каждую из оставшихся СУБД более внимательно:



**PostgreSQL** – это объектно-реляционная система управления базами данных (ОПСУБД, ORDBMS), основанная на POSTGRES, Version 4.2 — программе, разработанной на факультете компьютерных наук Калифорнийского университета в Беркли. В POSTGRES появилось множество новшеств, которые были реализованы в некоторых коммерческих СУБД гораздо позднее. [21]

**MySQL** – одна из самых популярных баз данных для веб-приложений. Фактически, является стандартом de facto для веб-серверов, которые работают под управлением операционной системы Linux. MySQL – это бесплатный пакет программ, однако новые версии выходят постоянно, расширяя функционал и улучшая безопасность. Существуют специальные платные версии, предназначенные для коммерческого использования. В бесплатной версии наибольший упор делается на скорость и надежность, а не на полноту функционала, который может стать и достоинством, и недостатком - в зависимости от области внедрения. [5]

#### Достоинства:

- Распространяется как в бесплатной версии, так и в платной;
- Хороший функционал даже в бесплатной версии;
- Поддерживает набор пользовательских интерфейсов;
- Может работать с другими базами данных, включая DB2 и Oracle;
- Обеспечивает высокую защиту данных.

#### Недостатки:

- Требуется много времени на настройку выполнения даже простых задач;
- Отсутствует поддержка XML или OLAP;
- Для бесплатной версии доступна только платная поддержка.

**MongoDB** – кроссплатформенная документно-ориентированная система управления БД. Классифицированная как база данных NoSQL, MongoDB отходит от традиционных основ реляционной структуры базы данных в пользу

JSON-подобных документов с динамическими схемами (MongoDB называет этот формат BSON), что делает интеграцию данных в определенных видах приложений проще и быстрее. Базы данных NoSQL появились относительно недавно с целью обеспечить производительность, масштабируемость и гибкость, необходимые для современных приложений. [6]

Подводя итог, можно сразу отсеять **MongoDB**, поскольку для реализации были выбраны реляционные базы данных, и мы знаем, что другие модели данных не будут использоваться, а данная СУБД изначально не была предназначена для обработки подобных моделей данных. Хотя MongoDB и может выполнять операции с SQL-запросами, однако, в данном случае необходимо производить дополнительную настройку, кроме того могут возникнуть серьёзные проблемы с производительностью, что неприемлемо в нашем случае. **PostgreSQL**, несмотря на все свои преимущества, обладает и критичными недостатками, такими как низкая производительность и длительность развёртки. Поскольку мы планируем использование достаточно небольшой базы данных, над которой не будет необходимости совершать сложные действия с миллионами объектов или одновременно работать с различными типами данных, то логичнее будет отдать предпочтение более скоростным СУБД, пожертвовав функционалом, который, вероятно, даже не будет использоваться в полной мере. **MySQL**, хоть и имеет некоторые недостатки, всё же позволяет достаточно быстро работать с выбранным типом данных, обеспечивая широкий функционал, кроме того, основными принципами данной СУБД являются надежность работы и безопасность данных, так же данная система выделяется наиболее дружелюбным к пользователю интерфейсом, таким образом преимущества перевешивают недостатки и именно по этой причине **MySQL** была выбрана для использования.

Выбрав СУБД стоит определиться с интерфейсом, при помощи которого непосредственный пользователь будет взаимодействовать с базами данных. Поскольку была выбрана MySQL, которая имеет клиент-серверную архитектуру и ориентирована прежде всего на web-приложения, имеет смысл использовать именно web-интерфейс для взаимодействия пользователя с базой данной. Кроме того, он имеет несколько преимуществ перед локальным приложением, устанавливаемым на рабочую станцию пользователя:

- Портативность – работа с системой через web-интерфейс требует только наличия Internet соединения с сервером, на котором расположен MySQL.
- Простота создания – создание web-интерфейса гораздо быстрее и проще, а, следовательно, дешевле, поскольку позволяет использовать множество уже готовых решений, ускоряющих разработку, например, оформление интерфейса при помощи стилей CSS.
- Ориентированность на данный тип интерфейса – как уже было сказано, MySQL обладает клиент-серверной архитектурой, таким образом, использование локального приложения в любом случае требовало бы создания и развертывания сервера для работы СУБД.

Таким образом, для дальнейшей разработки было выбрано приложение с web-интерфейсом.

Первым, что пользователь увидит, воспользовавшись системой, станет, несомненно, его интерфейс, он же и определяет насколько просто, быстро и удобно пользоваться данной системой. Именно по этой причине, мы уделим внимание некоторым основным принципам разработки web-интерфейса, которые являются ключевыми:

- Естественность (интуитивность).

Интерфейс не должен вызывать у пользователя сложностей в поиске необходимых элементов для управления процессом решения задачи.

- Непротиворечивость (идентичность).

Интерфейс не должен различаться в частях системы, должен использовать одинаковые методы и приёмы для работы с ним, а также работа с системой через интерфейс должна соответствовать установленным, привычным нормам.

- Не избыточность.

Пользователь должен вводить только минимальную информацию для работы или управления системой. Аналогично, не стоит требовать от пользователя ввести информацию, которая была введена ранее или которая может быть автоматически получена из системы.

- Доступ к системе помощи.

В процессе работы необходимо, чтобы система обеспечивала пользователя необходимыми инструкциями и помощью в решении проблем. Кроме того, сообщения об ошибках должны быть полезны, просты и понятны пользователю, обычно, отвечая всего на три важных вопроса:

1. в чем заключается проблема?
2. как исправить эту проблему сейчас?
3. как сделать так, чтобы проблема не повторилась?

- Гибкость.

Интерфейс системы должен обслуживать пользователя с различными уровнями подготовки.

Поскольку наша система включает web-интерфейс, то необходимой частью работы будет использование языка разметки документов HTML для создания web-страниц.

**HTML** (HyperText Markup Language – «язык гипертекстовой разметки») – стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML. Данный язык интерпретируется браузерами; полученный в результате

интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства. [1]

Он представляет собой достаточно простой, но крайне функциональный набор различных элементов (т.н. «тэгов»), которые описывают структуру документа и позволяет выделить в тексте отдельные логические части, поместить на web-страницу различные мультимедийные элементы, организовать ссылки для связи с другими документами и т.д.

Кроме того, web-страницы необходимо будет оформить, чтобы пользователь мог хоть как-то ориентироваться в различных элементах страниц. Для этой цели можно настраивать оформление каждой страницы отдельно в каждом из HTML файлов или же использовать каскадные таблицы стилей, или CSS.

**CSS** – это простой язык дизайна, предназначенный для упрощения процесса представления веб-страниц. CSS обрабатывает внешний вид веб-страницы. Используя CSS, можно контролировать цвет текста, стиль шрифтов, расстояние между абзацами, размер и расположение столбцов, фоновые изображения, макеты, варианты отображения для разных устройств и размеры экрана, а также множество других эффектов. CSS обеспечивает мощный контроль над представлением HTML-документа. [26]

Выбор языка разработки web-части системы является также важной задачей. Его функционал должен наиболее оптимально соответствовать задачам, что позволит в дальнейшем избавиться от различных проблем, вызванных ограничениями и особенностями различных языков.

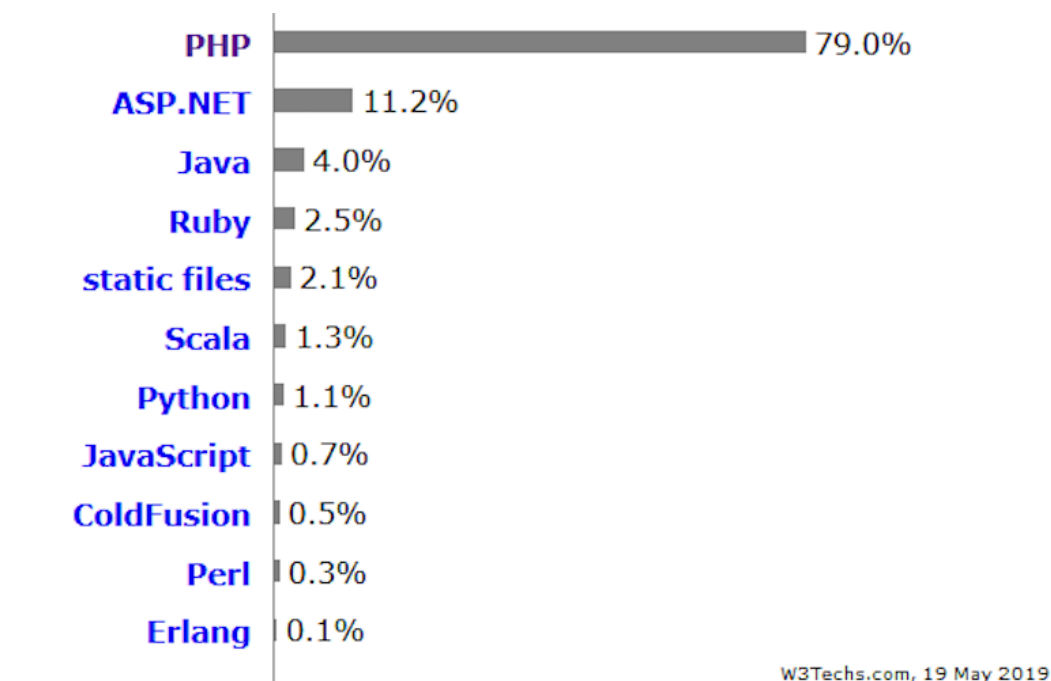


Рис. 1 Наиболее популярные языки web-программирования

[Интернет-ресурс: [https://skillbox.ru/media/code/chto\\_vybrat\\_asp\\_net\\_ili\\_php/](https://skillbox.ru/media/code/chto_vybrat_asp_net_ili_php/)]

Как видно из Рис. 1, наиболее популярными языками web-разработки являются PHP и ASP.NET. Хотя PHP побеждает с большим отрывом, всё же проведём их сравнение.

Таблица 1.  
Сравнение языков программирования

Параметры	PHP 5.6	ASP.NET
Цена	бесплатно	бесплатно
Скорость	высокая	средняя
Эффективность	высокая	низкая
Безопасность	высокая	высокая
Платформа (стабильность)	стабильная	нестабильная (за счет IIS)
Платформа для работы	любая	win32 (только на IIS)
Доступность исходников	да	нет
Обработка исключений	да	да
Поддержка ООП	хорошо	хорошо

Проанализировав результаты, приведённые в Таблица 1, преимущества языка PHP становятся очевидны, поэтому наиболее оптимальным решением будет использовать именно его.

Кроме того, для работы любого web-ресурса необходим некий локальный сервер, обрабатывающий PHP и SQL запросы. На данный момент наиболее популярными и распространёнными готовыми серверами являются Denwer, OpenServer и XAMPP, которые и были рассмотрены.

**Denwer:** Сразу после установки доступен полностью работающий веб-сервер Apache, интерпретатор PHP, СУБД MySQL, панель phpMyAdmin и система управления виртуальными хостами.

Однако, Denwer длительное время не обновлялся и поддерживает PHP только до версии 5, что является большим недостатком.

**XAMPP:** кроссплатформенная сборка веб-сервера, содержащая Apache, MySQL, интерпретатор скриптов PHP, язык программирования Perl и большое количество дополнительных библиотек, позволяющих запустить полноценный веб-сервер прямо «из коробки».

Однако у XAMPP есть существенный недостаток. Несмотря на кроссплатформенность, простоту и достаточный набор инструментария «из коробки», он работает только с 32х разрядными версиями Windows, чем ограничивает своё применение на данной платформе.

**OpenServer:** Портативный локальный WAMP/WNMP сервер, имеющий многофункциональную управляющую программу и большой выбор подключаемых компонентов.

Для отладки скриптов в различном окружении Open Server предлагает на выбор сразу два вида HTTP серверов, различные версии PHP и СУБД модулей, а также возможность быстрого переключения между ними.

#### HTTP модули:

- Apache 2.2.21;

- Nginx 1.0.11.

СУБД модули:

- MySQL 5.1.61;
- MySQL 5.5.20;
- PostgreSQL 9.1.1.

PHP модули:

- PHP версии 5.x
- PHP версии 7.x

Отличный набор инструментов:

- HeidiSQL
- Adminer
- PHPMyAdmin
- PHPPgAdmin
- PgAdmin
- Perl
- FTP сервер
- Sendmail
- Memcached сервер.

Реализовано удобное подключение правил rewrite через файлы .htaccess в корне домена, а PHP работает в режиме True FastCGI. Все компоненты взяты с официальных репозиторий и регулярно обновляются до актуальных версий. Кроме того, функционал OS включает в себя весь набор функций Denwer, но с обновлениями, большим функционалом и дружелюбным интерфейсом, что ещё сильнее снижает возможности применения Denwer.

На основе приведенных выше аргументов в качестве сервера разработки был выбран OpenServer.

Важной составляющей СУБД является интерфейс её администрирования. В выбранном OpenServer эту роль выполняет phpMyAdmin, который является



простым и удобным web-интерфейсом, позволяющий осуществлять администрирование MySQL сервера прямо в окне браузера, выполнять SQL команды, просматривать, изменять и добавлять содержимое таблиц и баз данных даже без непосредственного ввода SQL команд, посредством заполнения удобных форм, самостоятельно генерирующих необходимые запросы. [3]

Кроме того, неотъемлемой частью любой информационной системы являются базы данных, которые необходимо спроектировать перед началом разработки. Для разработки проекта базы данных информационной системы было использовано приложение DBDesigner, которое позволяет легко создать необходимую структуру из таблиц, полей и связей между полями при помощи визуального программирования, а затем экспортировать полученный результат в формате SQL запросов или же в виде .sql файла для последующего импорта в любую СУБД, поддерживающую работу с SQL. [2]

Крайне полезным элементом разработки информационной системы является составление диаграмм UML, которые являются языком графического описания абстрактной системы. Различные диаграммы позволяют детально и понятно отразить элементы системы, её структуру, внутренние процессы, поведение и так далее, что в дальнейшем позволяет значительно ускорить процесс реализации и позволяет избежать неточностей понимания тех или иных элементов или процессов системы. [3] Для создания UML диаграмм было использовано web-приложение Draw.io, которое обладает всем необходимым инструментарием, большим набором шаблонов для быстрого старта, наборами элементов для любого типа диаграмм, поддерживает функции сохранения и загрузки диаграмм в файл и помимо всего прочего может быть запущено на любой операционной системе.

### **1.3 Формализованное описание технического задания**

## ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на разработку информационной системы

«Информационная система по управлению материальной базой предприятия»

Составлен на основе ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»

### 1. Общие сведения.

#### 1.1. Название организации-заказчика:

ФГБОУ ВО «УрГПУ»

#### 1.2. Название продукта разработки:

Информационная система по управлению материальной базой  
предприятия

#### 1.3. Назначение продукта:

Автоматизация, упрощение и/или стандартизация управления  
материальными базами для различных предприятий

#### 1.4. Плановые сроки начала и окончания работ:

В соответствии с планом выполнения ВКР (01.09.2019 – 19.05.2020)

### 2. Характеристика области применения продукта.

#### 2.1. Процессы и структуры, в которых предполагается использование продукта разработки:

Управление материальной базой любого предприятия

#### 2.2. Характеристика персонала (количество, квалификация, степень готовности):

2.2.1. Разработчик – знание SQL, PHP и JavaScript, умение разработки  
web-интерфейсов;

2.2.2. Администратор – опыт работы с базами данных на SQL;

2.2.3. Пользователь – уверенный пользователь ПК.

### 3. Требования к продукту разработки.

#### 3.1. Требования к продукту в целом:

Возможность управления материальной базой.

- 3.2. Аппаратные требования:

Персональный компьютер с возможностью выхода в Интернет.
- 3.3. Указание системного программного обеспечения (операционные системы, браузеры, программные платформы и т.п.):

Любая операционная система, имеющая возможность установки графического браузера с поддержкой HTML5
- 3.4. Указание программного обеспечения, используемого для реализации:

OpenServer, NotePad++
- 3.5. Форматы входных и выходных данных:

В качестве входных данных пользователь заполняет форму с помощью клавиатуры. На выходе пользователь получает базу данных.
- 3.6. Источники данных и порядок их ввода в систему, порядок вывода, хранения:

Источником данных выступает информация, вводимая пользователем с клавиатуры, вывод и хранение осуществляется при помощи базы данных.
- 3.7. Порядок взаимодействия с другими системами, возможности обмена информацией:

Возможность получать информацию из баз данных системы
- 3.8. Меры защиты информации:

Не предусмотрены
- 4. Требования к пользовательскому интерфейсу.
  - 4.1. Навигация по сайту через блок меню, форма вывода данных;
  - 4.2. Перемещение по сайту, вход, выход;
  - 4.3. Проверка на ошибки данных формы.
- 5. Перечень сопроводительной документации:

Руководство пользователя.

6. Порядок сдачи-приемки продукта.

В соответствии с планом выполнения ВКР.

## Глава 2. Разработка информационной системы по управлению материальной базой предприятия

### 2.1 Модельные представления разработанной информационной системы

Перед разработкой системы для упрощения дальнейшей работы были созданы различные UML диаграммы и информационная модель данного проекта.

На Рис. 2 представлена UML диаграмма использования.

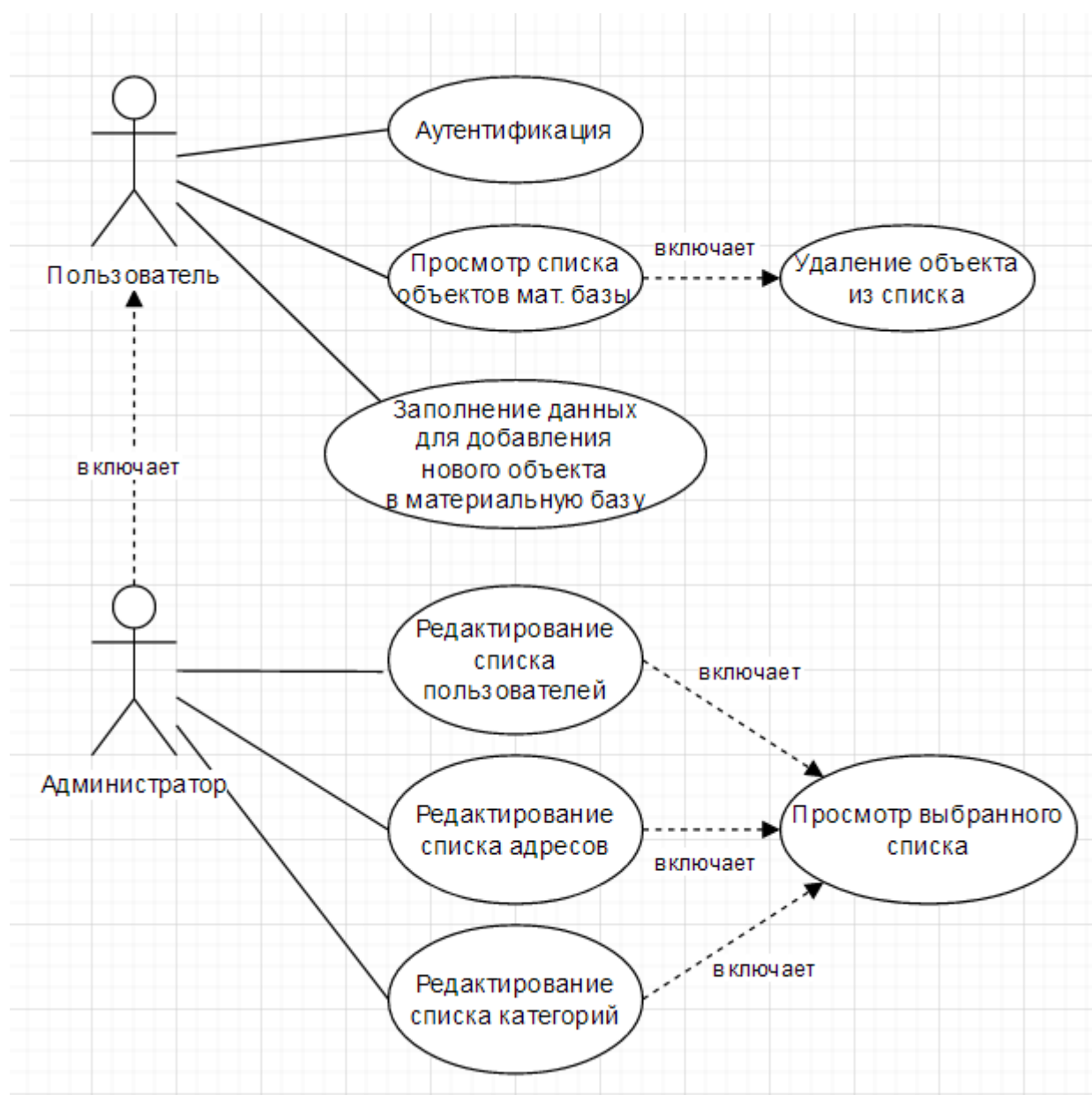


Рис. 2 Диаграмма использования

На данной диаграмме изображены возможности пользователя и администратора системы.

Пользователь имеет права на аутентификацию, просмотр списка всех объектов, находящихся в материальной база, а также может добавить новый объект, заполнив необходимую форму.

Администратор имеет те же права, что и пользователь, но сверх того может просматривать и редактировать списки пользователей, адресов расположения материальной базы, а также категорий, по которым объекты могут быть отсортированы.

Для упрощения разработки алгоритмической и логической реализации выполняемой системой работы была создана диаграмма деятельности, изображенная на Рис. 3

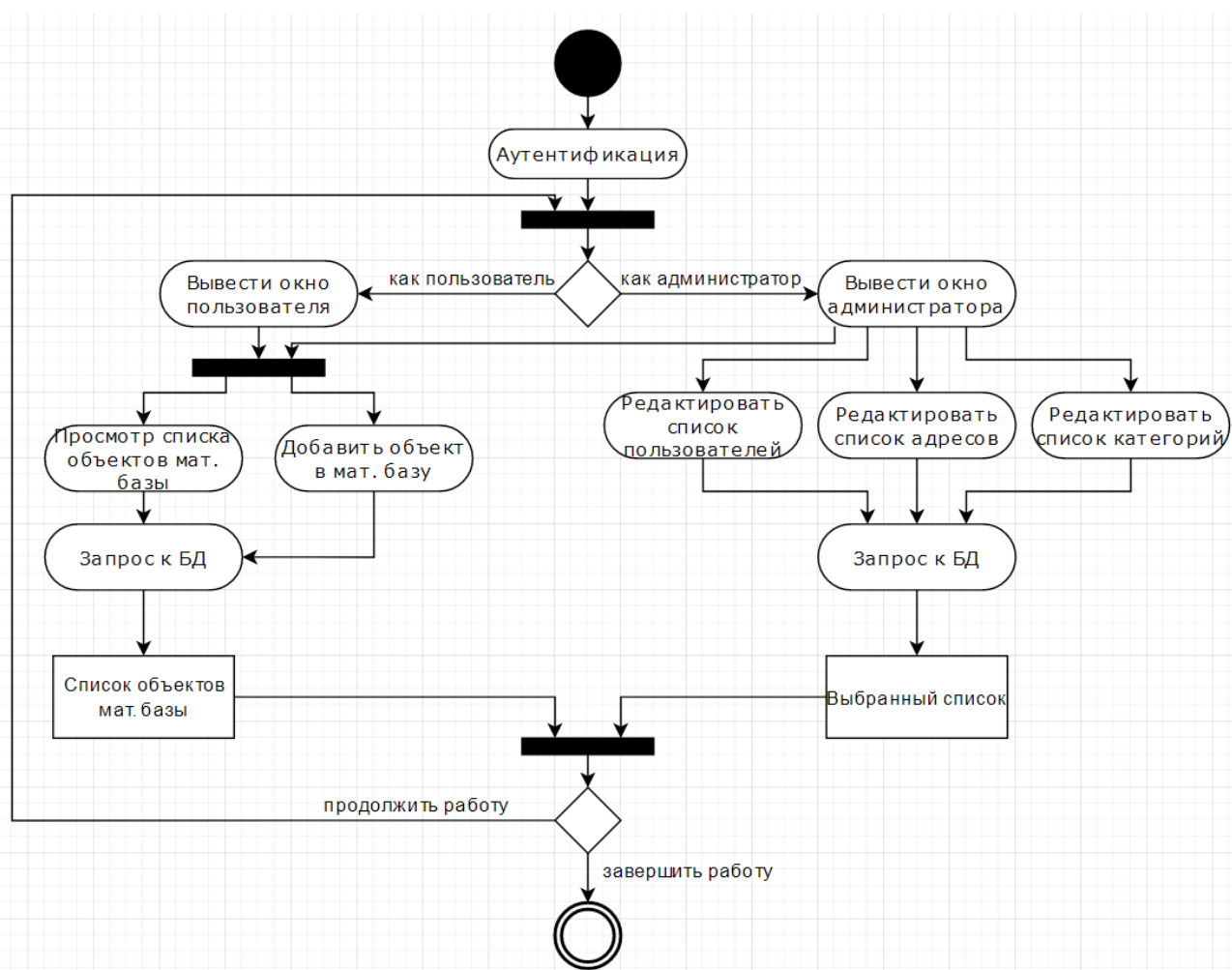


Рис. 3 Диаграмма деятельности

Работа в системе выполняется по следующему алгоритму:

1. Пользователь входит в систему и проходит авторизацию. Система аутентифицирует пользователя либо как обычного пользователя, либо как пользователя с правами администратора и выводит соответствующее рабочее окно.
2. Пользователь выбирает действие, которое он хочет совершить, что вызывает запрос к базе данных. Результат обработки (вывод текущей информации в разделе или добавление новой) выводится пользователю.
3. Пользователь может либо закончить работу, выйдя из системы, либо продолжить работу, в таком случае, его снова вернёт в начальное рабочее окно пользователя, где алгоритм будет повторяться.

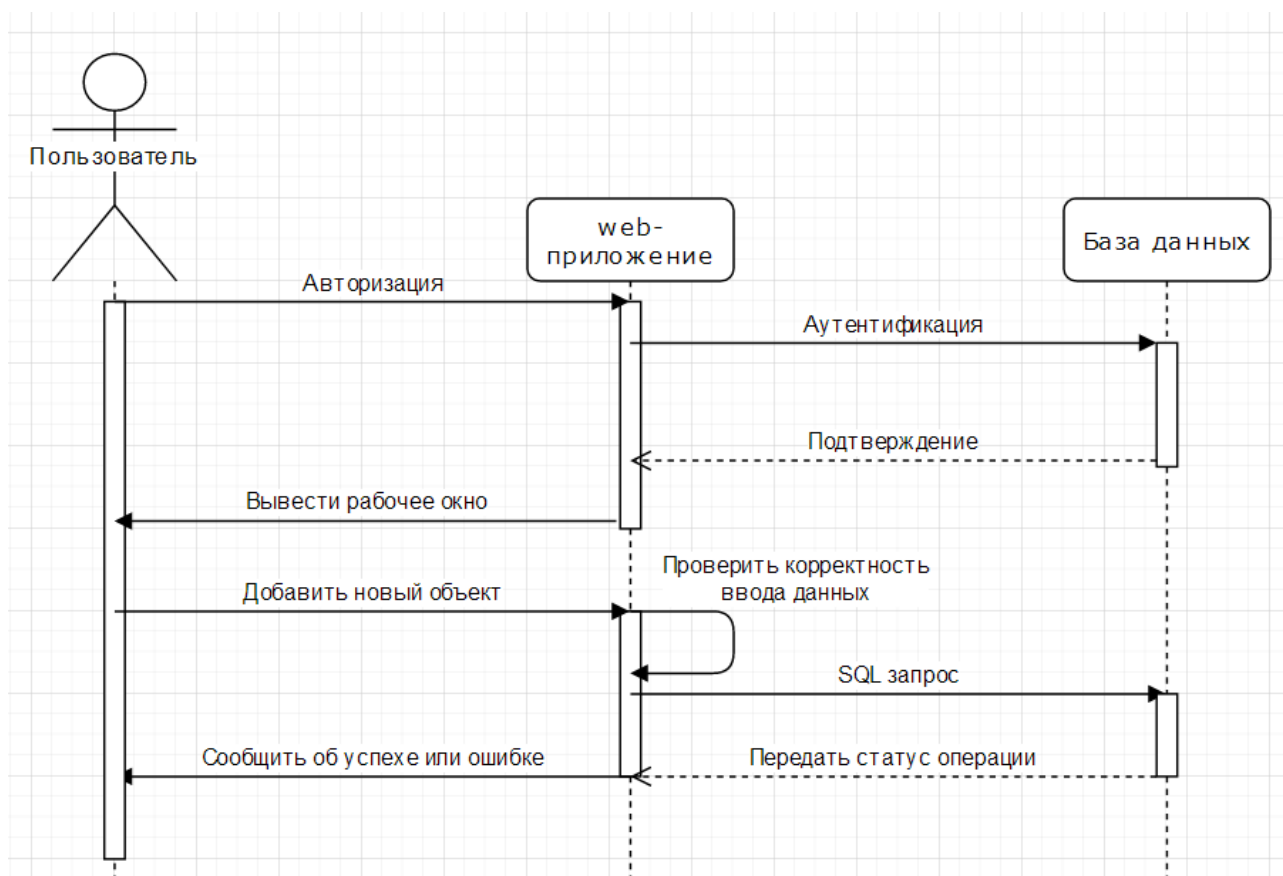


Рис. 4 Диаграмма последовательности добавления нового объекта мат. базы

На Рис. 4 представлена диаграмма последовательности для добавления нового объекта в материальную базу. Пользователь должен авторизоваться, после чего, в случае успешной аутентификации, ему будет выведено рабочее

окно, где он может внести необходимые данные об объекте и попытаться его добавить в материальную базу.

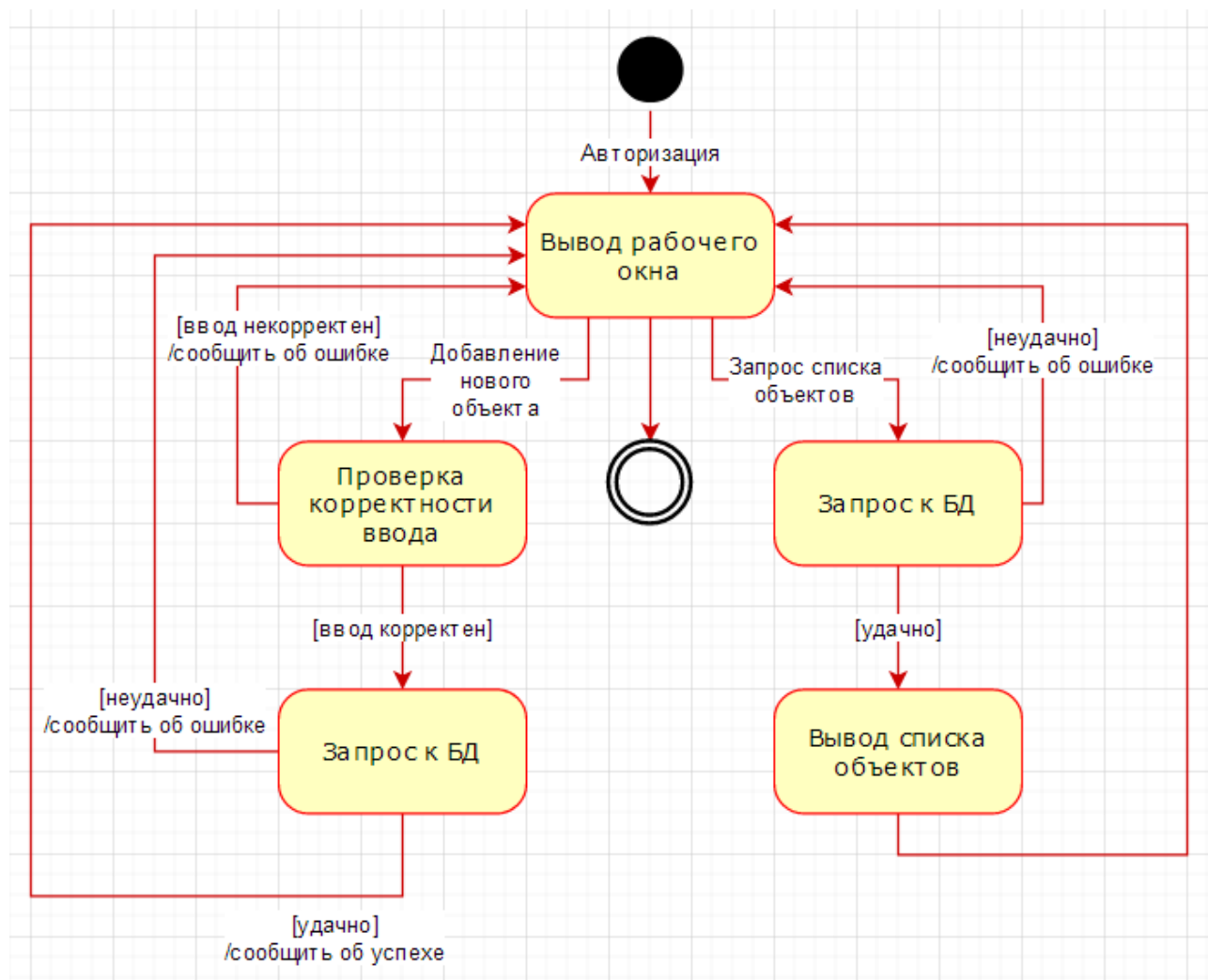


Рис. 5 Диаграмма состояний web-интерфейса

На Рис. 5 приведена диаграмма состояний web-интерфейса системы при работе с пользователем. После авторизации выводится рабочее окно и интерфейс ожидает, что пользователь совершит одно из действий: добавит новый объект, запросит текущий список объектов или же завершит работу. Диаграмма состояний для администратора будет схожей, с различием только в возможности вывести несколько других типов списков.

Следующим шагом была разработка функциональной модели системы. Функциональная модель позволяет отобразить функциональную структуру системы, т.е. производимые ей действия и связи между различными



элементами при помощи этих действий. Функциональная модель системы первого уровня изображена на Рис. 6



Рис. 6 Функциональная модель первого уровня

В качестве входа в основной блок выступает информация из базы данных о всех объектах материальной базы, включая пользователей, на выход система выводит новую, изменённую базу данных о материальных ресурсах, которую и видит пользователь через web-интерфейс. Механизмами являются программное обеспечение, такое как браузер, и пользователь. Средствами управления выступают нормативные документы, такие как, например, ГОСТ построения баз данных, и требования для разработки web-страниц, поскольку система использует web-сайт в качестве интерфейса пользователя.

Далее рассмотрим функциональную модель системы второго уровня, приведенную на Рис. 7 .

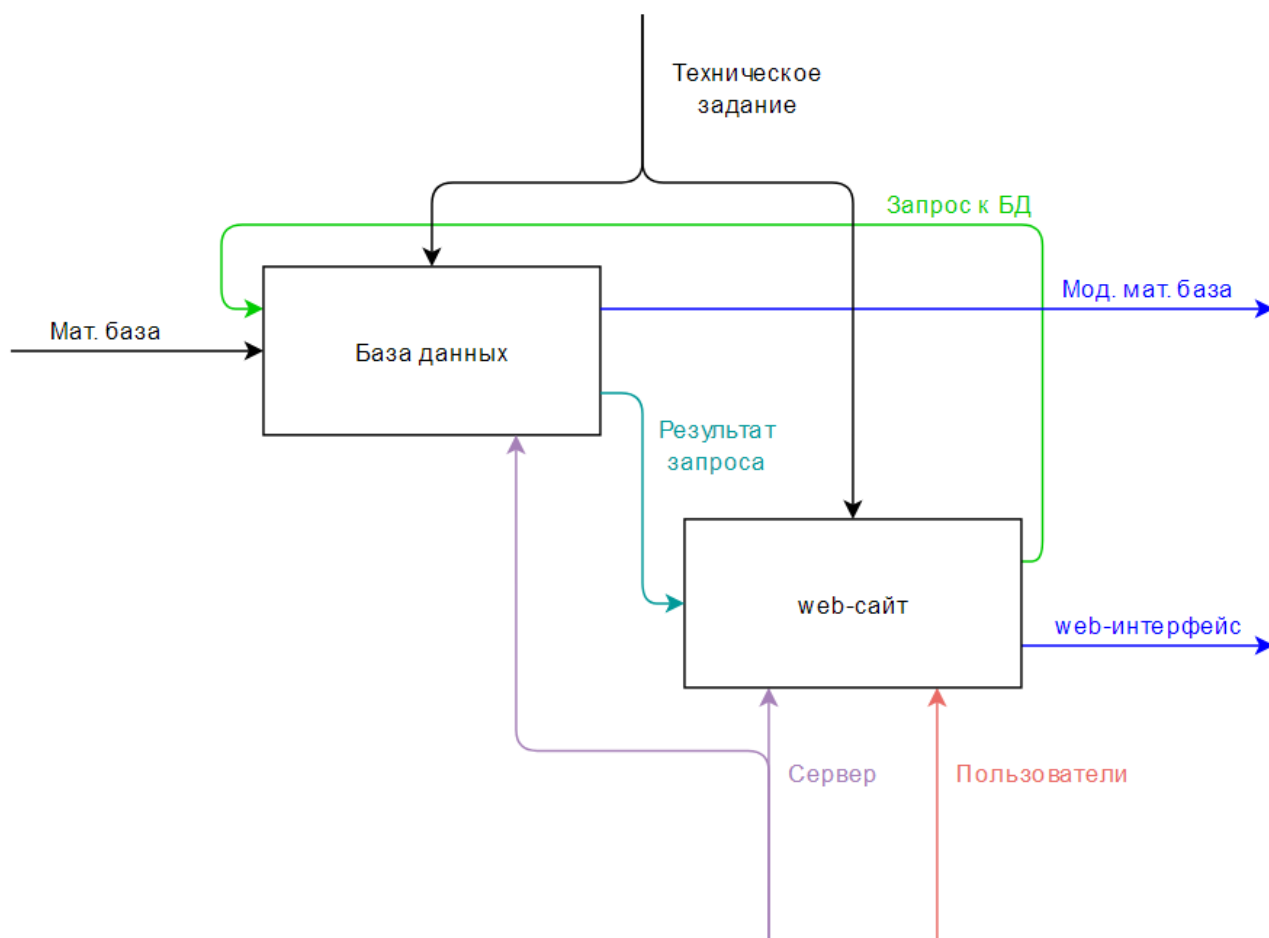


Рис. 7 Функциональная модель второго уровня

На данной модели представлены два блока: База данных и web-сайт, выступающий в роли интерфейса пользователя. Пользователь посылает запрос к базе данных, на добавление, изменение или поиск нужной вещи в материальной базе, далее база данных обрабатывает запрос и посылает web-приложению необходимые для пользователя данные, выводимые на экран, или успешно редактирует или добавляет новые данные в базу.

Кроме того, проведем декомпозицию блока web-сайт, которая представлена на Рис. 8

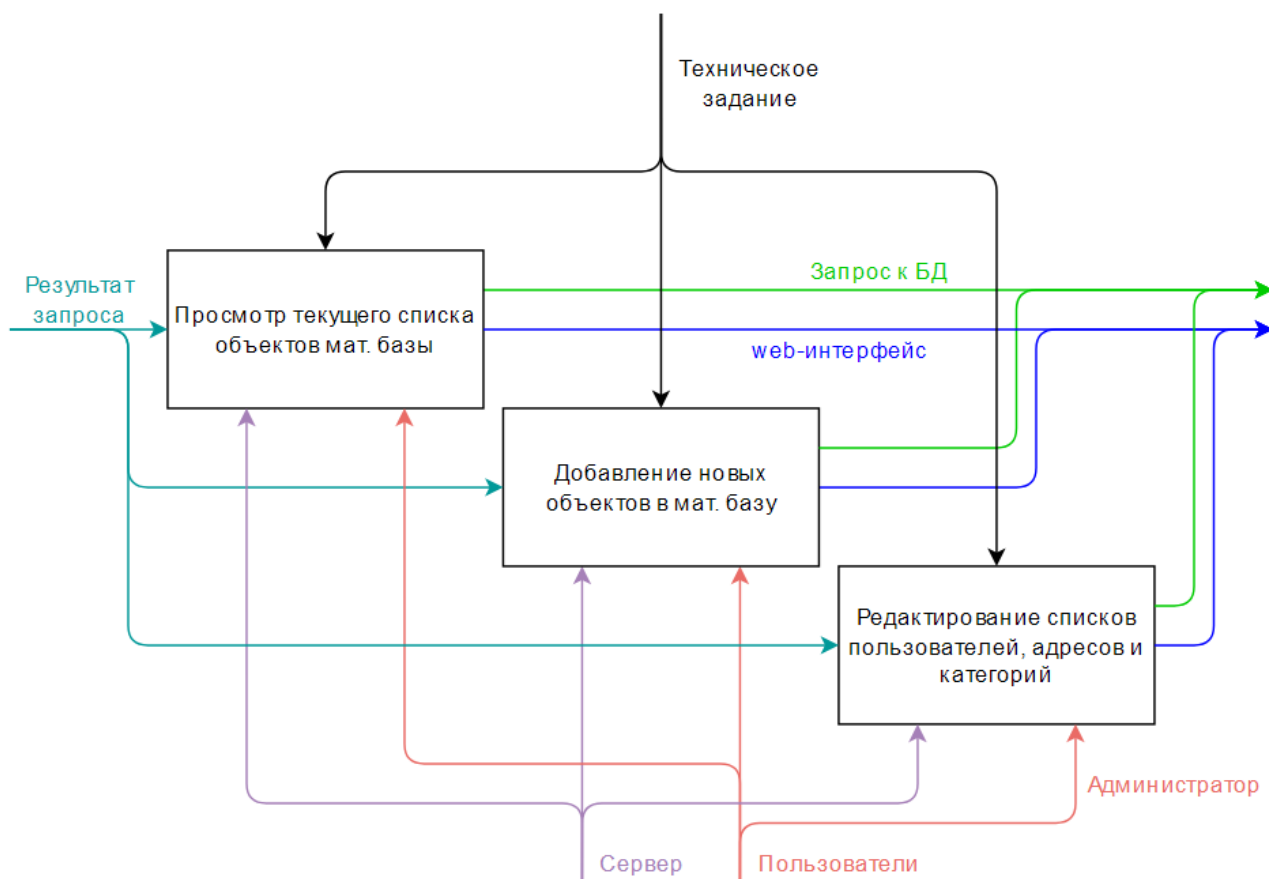


Рис. 8 Декомпозиция блока web-сайт

Следующим шагом будет проектирование самой базы данных, которая будет использоваться системой. Разработка проекта базы данных осуществлялась в онлайн сервисе DBDesigner.

Спроектированная база данных, которую можно увидеть на Рис. 9 , состоит из 5 связанных между собой таблиц:

1. «Объекты», атрибутами которой являются: obj\_id, cat\_id, rasp\_id, otv\_id, obj\_name, obj\_comment;
2. «Категории», атрибутами которой являются: cat\_id и cat\_name;
3. «Расположения», атрибутами которой являются: rasp\_id и rasp\_name;
4. «Ответственные», атрибутами которой являются: otv\_id, otv\_name, otv\_surname, otv\_patron, otv\_comment;
5. «Пользователи», атрибутами которой являются: user\_id, user\_login, user\_pass.

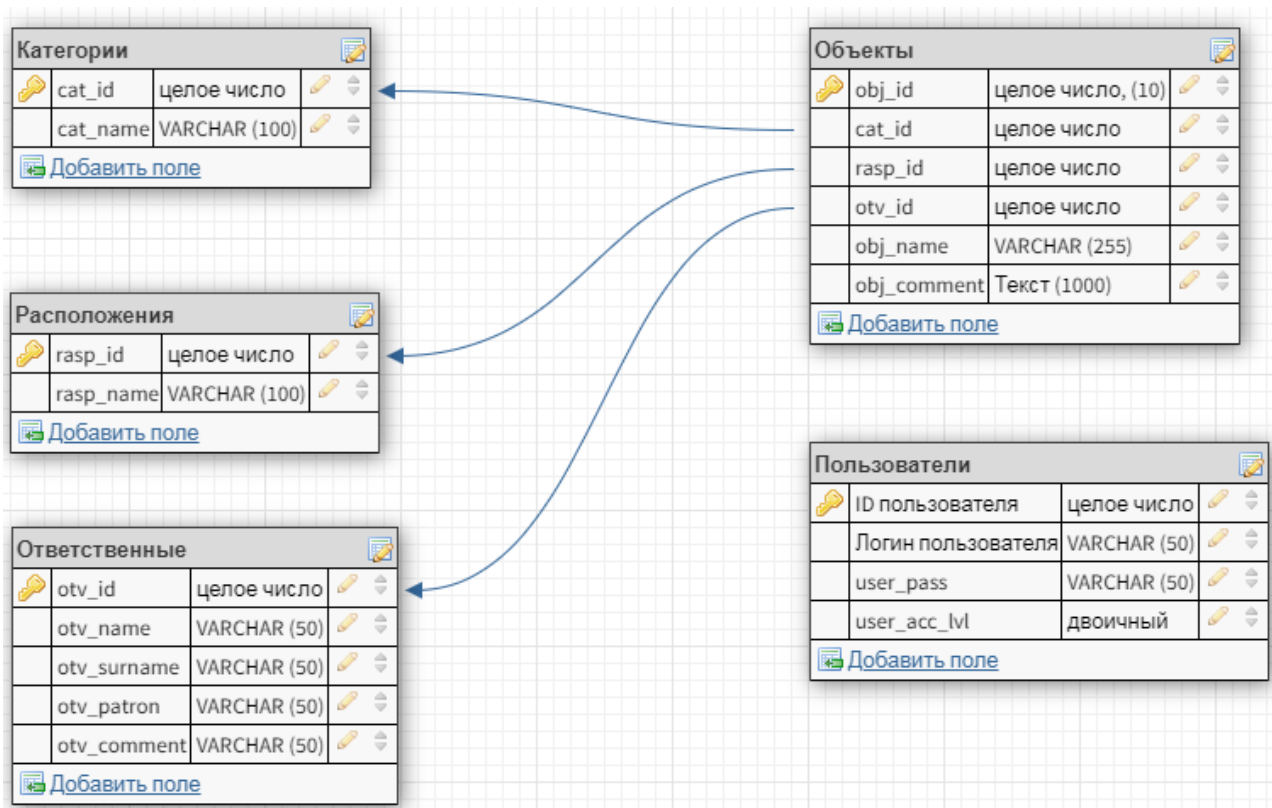


Рис. 9 Таблицы базы данных в DBDesigner

После создания базы данных в DBDesigner её можно экспортировать в виде готового SQL запроса для создания базы данных в любой среде, например, в PhpMyAdmin, входящий в состав выбранного ранее OpenServer.

Результат экспорта из DBDesigner представлен на Рис. 10

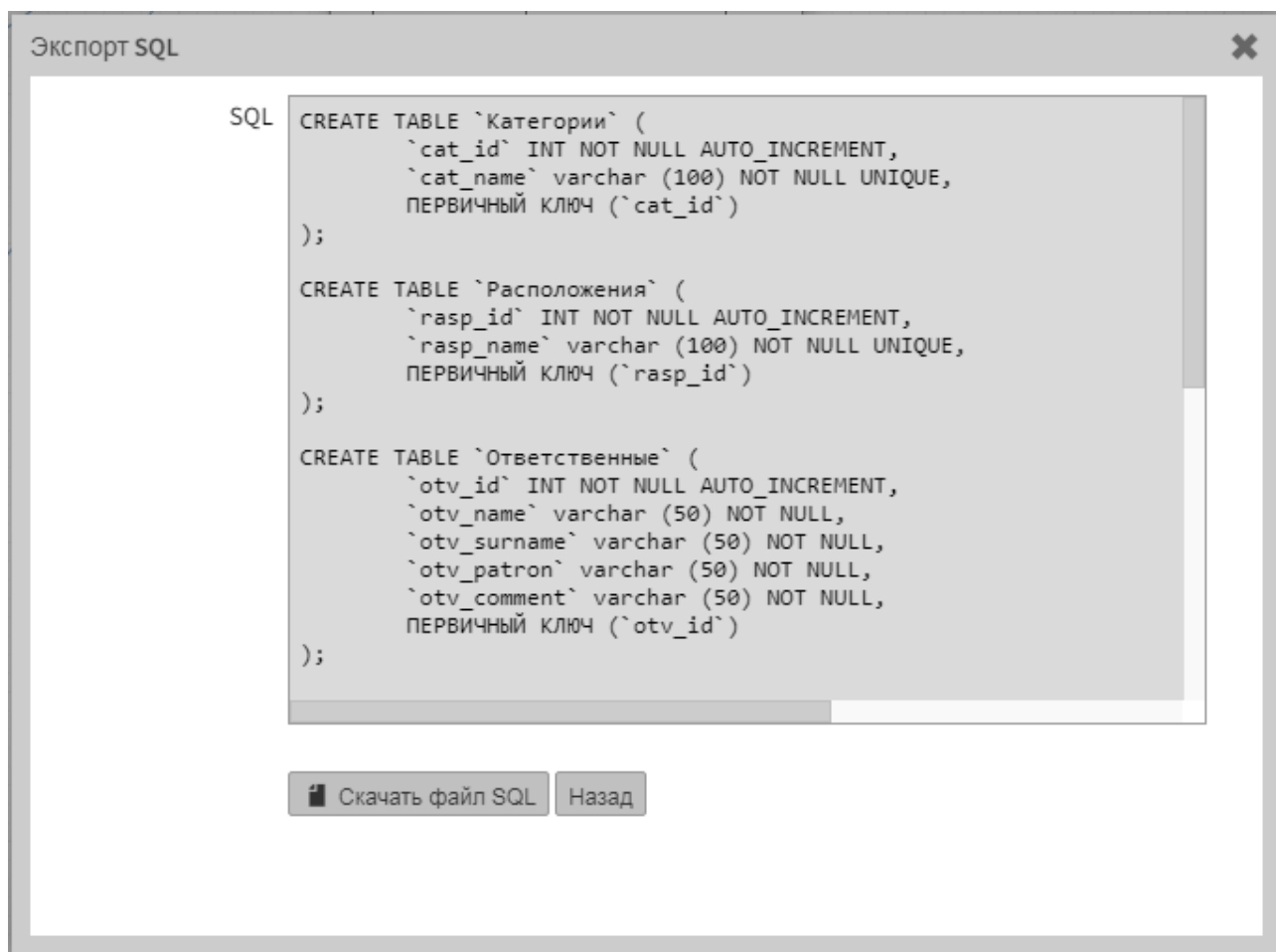


Рис. 10 Окно экспорта SQL кода из DBDesigner

Полученный код мы можем либо просто скопировать, либо сохранить в формате .sql файла для дальнейшего импорта.

Далее, в phpMyAdmin необходимо создать пустую базу данных, а затем воспользоваться либо выполнением скопированного из DBDesigner SQL запроса, либо, если база была сохранена как .sql файл, можно воспользоваться функцией импорта из файла. Это позволит просто импортировать созданную ранее структуру базы данных.

Воспользуемся функцией выполнения SQL запроса, что продемонстрировано на Рис. 11 , результат выполнения операции продемонстрирован на Рис. 12

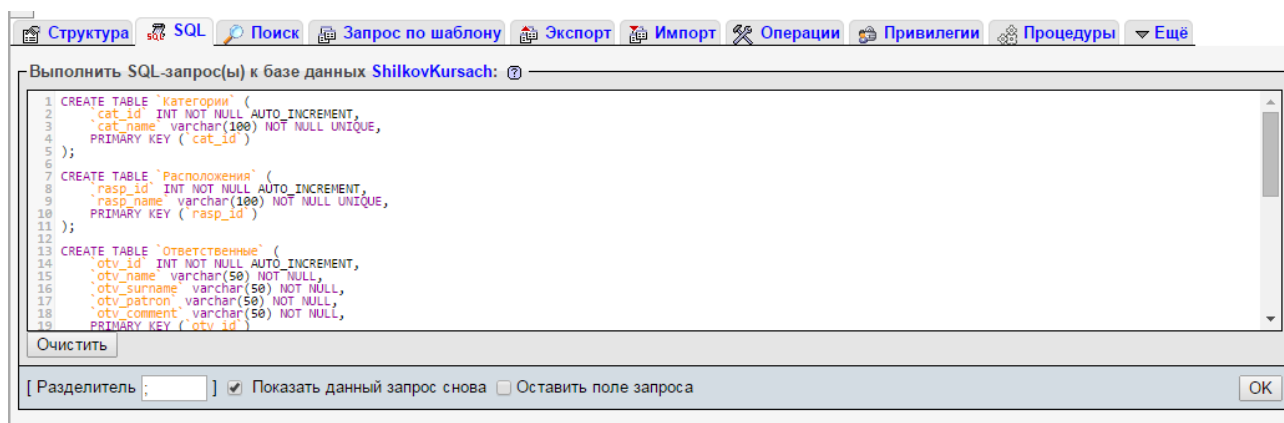


Рис. 11 Создание структуры БД при помощи SQL запроса

<div>СтруктураSQLПоискЗапрос по шаблонуЭкспортИмпортОперацииПривилегии</div>											
	Таблица ▲	Действие					Строки ?	Тип	Сравнение	Размер	Фрагментировано
<input type="checkbox"/>	Категории						~0	InnoDB	utf8_general_ci	32 КБ	-
<input type="checkbox"/>	Объекты						~0	InnoDB	utf8_general_ci	64 КБ	-
<input type="checkbox"/>	Ответственные						~0	InnoDB	utf8_general_ci	16 КБ	-
<input type="checkbox"/>	Пользователи						~0	InnoDB	utf8_general_ci	32 КБ	-
<input type="checkbox"/>	Расположения						~0	InnoDB	utf8_general_ci	32 КБ	-
	5 таблиц	Всего					0	InnoDB	utf8_general_ci	176 КБ	0 байт

Рис. 12 Структура БД после импорта

Осталось вручную создать пользователя или пользователей с правами администратора, чтобы начать работу. Кроме того, можно сразу создать и обычного пользователя, хотя это будет возможно сделать и через интерфейс самой системы. Примерный результат показан на Рис. 13

		user_id	user_login	user_pass	user_acc_lvl
<input type="checkbox"/>		1	user	user	0
<input type="checkbox"/>		2	admin	admin	1

Рис. 13 Таблица Пользователи

После создания базы данных необходимо реализовать пользовательский интерфейс, который будет с ней взаимодействовать. Для начала необходимо создать окно логина, которое будет являться первым, что увидит пользователь. Результат работы показан на Рис. 14 и Рис. 15

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE></TITLE>
<link href="style.css?2" rel="stylesheet"/>
<script src="scr/jquery.min.js"></script>
</HEAD>
<BODY>
<section class="wrap">
<H1>Войдите в систему</H1>
<form method="POST" id="login" action="" >
<div class="wrap_l">
<input type="text" class="login" name="login" placeholder="Логин" required minlength="2" maxlength="30"/>
<input type="text" class="pass" name="pass" placeholder="Пароль" required minlength="2" maxlength="30"/>
</div>
<button id="men_submit" class="subm_but" style="width: 300px; height: 30px; margin-top: 5px;">Войти</button>
<?php if(isset($bad)) echo "<div class=\"top_btn\" id=\"bad\" >Неверные данные!</div>"?>
</form>
</section>
</BODY>
</HTML>

```

Рис. 14 HTML часть главной страницы

```

<?php
ini_set('display_errors','Off');

session_set_cookie_params(90000);
session_start();
if(isset($_SESSION['cool'])){
header("Location: ./");
exit;
}

if(isset($_POST['login']) && isset($_POST['pass'])){

require_once 'scr/db_connect.php';
$link = mysqli_real_escape_string($link, htmlentities($_POST['login']));
$pass = mysqli_real_escape_string($link, htmlentities($_POST['pass']));

$query = "SELECT * FROM `пользователи` WHERE `user_login` = '". $login.'" and `user_pass` = '". $pass.'" ";
if (!$result = $link->query($query)) {
/*echo "Ошибка sql: \n";
echo "Запрос: " . $query . "\n";
echo "Ошибка: " . $link->errno . " " . $link->error . "\n";*/
echo 'error';
exit;
}

if ($result->num_rows == 1) {
$resultArray = $result->fetch_assoc();
$_SESSION['cool'] = $_POST['login'];
$_SESSION['acc_lvl'] = $resultArray["user_acc_lvl"];
header("Location: ./");
$result->free();
$link->close();
exit;
}

else $bad = true;
$result->free();
$link->close();
}

?>

```

Рис. 15 php часть главной страницы

Затем пользователю должно выводиться рабочее окно. Часть реализации представлена на Рис. 16 . Попадая на страницу, пользователь проходит

проверку на наличие прав и если он имеет права администратора, то ему выводятся дополнительные элементы интерфейса.

```
<HEAD>
<TITLE></TITLE>
<link href="style.css?2" rel="stylesheet"/>
<script src="scr/jquery.min.js"></script>
<script src="scr/jquery.cookie.js"></script>
</HEAD>
<BODY>
<section class="wrap">
<H1>Управление материальной базой предприятия</H1>
<p>Добро пожаловать, <?php echo $_SESSION['cool']; ?>! <a href="auth.php?do=logout">(выход)</a></p>
<hr/>
<b>Поиск по инвентарному номеру</b>
</br>
<input id="invnum" maxlength="10" size="40" placeholder="Введите инвентарный номер">
<input onclick="check();" type="button" value="Проверить">
</br>
</br>
<div id="obj_info"></div>
<H3>Объекты</H3>
<a href="lists.php?action=obj">Список объектов</a>
</br></br>
Добавить объект:
</br>
<input id="obj_name" maxlength="100" size="70" placeholder="Введите название объекта">
</br>
<select id="obj_cat">
<option value="none" hidden="">Выберите категорию</option>
<?php
$query ="SELECT * FROM `категории`;";
if (!$result = $link->query($query)) {
    echo '<option>error</option>';
    exit;
}
while ($row = $result->fetch_assoc()) {
    echo '<option value="'. $row["cat_id"].'">'. $row["cat_name"].'</option>';
}
?>
</select>
<select id="obj_rasp">
<option value="none" hidden="">Выберите расположение</option>
<?php
$query ="SELECT * FROM `расположения`;";
if (!$result = $link->query($query)) {
    echo '<option>error</option>';
    exit;
}
while ($row = $result->fetch_assoc()) {
    echo '<option value="'. $row["rasp_id"].'">'. $row["rasp_name"].'</option>';
}
?>
</select>
</br>
</br>
</div>
</div>
</BODY>
</HTML>
```

Рис. 16 Фрагмент реализации рабочего окна

Одним из действий, доступных пользователю, является просмотр текущего списка объектов материальной базы. Фрагмент реализации данной функции представлен на Рис. 17



```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE></TITLE>
<link href="style.css?2" rel="stylesheet"/>
<script src="scr/jquery.min.js"></script>
<script src="scr/jquery.cookie.js"></script>
</HEAD>
<BODY>
<div id="back" onclick="window.history.back()"></div>
<section class="wrap">
<?php
if (isset($_GET["action"])){
$action = $_GET["action"];

switch ($action){
case 'cat':
echo '<H1>Список категорий</H1><br/><hr>';
$query = "SELECT * FROM 'категории'";
if (!$result = $link->query($query)) {
echo 'error';
exit;
}
while ($row = $result->fetch_assoc()) {
echo $row["cat_id"].': <b>'. $row["cat_name"].'</b> <span class="pointer" onclick="del(\''. $row["cat_id"].'\',\'категории\',\''. $row["cat_n
echo '<hr>';
}
echo '** Номера соответствуют ID в базе данных';
break;
case 'rasp':
echo '<H1>Список расположений</H1><br/><hr>';
$query = "SELECT * FROM 'Расположения'";
if (!$result = $link->query($query)) {
echo 'error';
exit;
}
while ($row = $result->fetch_assoc()) {
echo $row["rasp_id"].': <b>'. $row["rasp_name"].'</b> <span class="pointer" onclick="del(\''. $row["rasp_id"].'\',\'расположения\',\''. $row[
echo '<hr>';
}
echo '** Номера соответствуют ID в базе данных';
break;

```

Рис. 17 Фрагмент реализации просмотра содержания материальной базы

Кроме того, в данном блоке по тому же принципу реализован просмотр списка пользователей, категорий, адресов и ответственных лиц для администратора.

Кроме того, если пользователь имеет инвентарный номер, он может проверить есть ли предмет с таким номером в материальной базе и получить информацию о нём, если такой предмет действительно есть. Реализация данной функции представлена на Рис. 18

```

<?php

ini_set("display_errors",1);
error_reporting(E_ALL);

require_once 'auth.php';
require_once 'scr/db_connect.php';

header("HTTP/1.1 200 OK");
http_response_code(201);
header("Status: 200 All rosy");

$resp = array();
$resp["status"] = 'empty';

$query = " SELECT `obj_id`,`obj_name`,`obj_comment`,`cat_name`,`otv_name`,`otv_surname`,`rasp_name`
FROM `Объекты`
LEFT JOIN `Категории` ON `Объекты`.`cat_id` = `Категории`.`cat_id`
LEFT JOIN `Ответственные` ON `Объекты`.`otv_id` = `Ответственные`.`otv_id`
LEFT JOIN `Расположения` ON `Объекты`.`rasp_id` = `Расположения`.`rasp_id`
WHERE obj_id = ".$_GET["id"].";";
if (!$result = $link->query($query)) {
    $resp["status"] = 'error';
}
while ($row = $result->fetch_assoc()) {
    $resp["obj_id"] = sprintf("%06d", $row["obj_id"]);
    $resp["obj_name"] = $row["obj_name"];
    $resp["obj_comment"] = $row["obj_comment"];
    $resp["cat_name"] = $row["cat_name"];
    $resp["rasp_name"] = $row["rasp_name"];
    $resp["otv_name"] = $row["otv_name"] . ' ' . $row["otv_surname"];
    $resp["can_del"] = $_SESSION['acc_lvl'];
    $resp["status"] = 'ok';
}

header('Content-type: application/json');
echo json_encode( $resp , JSON_UNESCAPED_UNICODE);

```

Рис. 18 Реализация проверки инвентарного номера

Далее, осталось реализовать функции добавления и удаления записей в таблицах базы данных. Результат представлен на Рис. 19 и Рис. 20

```

<?php

require_once 'auth.php';
require_once 'scr/db_connect.php';

$query = "DELETE FROM ".$_GET["table"]." WHERE `".$_GET["field"]."` = '".$_GET["id"]." ";
if (!$result = $link->query($query)) {
    echo "Ошибка sql: \n";
    echo "Запрос: " . $query . "\n";
    echo "Ошибка: " . $link->errno . " ".$link->error . "\n";
    echo 'error';
    exit;
}
?>

```

Рис. 19 Реализация удаления записи

```

<?php
header("HTTP/1.1 200 OK");
http_response_code(201);
header("Status: 200 All rosy");

require_once 'auth.php';
$res = '1';

if (isset($_GET["a"])){
    $action = $_GET["a"];
    require_once 'scr/db_connect.php';

    switch ($action){
        case 'cat':
            $query = "INSERT INTO `категории` (`cat_name`) VALUES ('".$_GET["cat"]."') ";
            if (!$result = $link->query($query)) $res = '0';
            break;
        case 'rasp':
            $query = "INSERT INTO `расположения` (`rasp_name`) VALUES ('".$_GET["rasp"]."') ";
            if (!$result = $link->query($query)) $res = '0';
            break;
        case 'otv':
            $query = "INSERT INTO `ответственные` (`otv_name`, `otv_surname`, `otv_comment`) VALUES ('".$_GET["name"]."','".$_GET["surname"]."','".$_GET["co
            if (!$result = $link->query($query)) $res = '0';
            break;
        case 'user':
            $query = "INSERT INTO `пользователи` (`user_login`, `user_pass`, `user_acc_lvl`) VALUES ('".$_GET["login"]."','".$_GET["pass"]."','".$_GET["acc_
            if (!$result = $link->query($query)) $res = '0';
            break;
        case 'obj':
            $query = "INSERT INTO `объекты` (`cat_id`, `rasp_id`, `otv_id`, `obj_name`, `obj_comment`) VALUES ('".$_GET["cat_id"]."','".$_GET["rasp_id"]."','".$_
            if (!$result = $link->query($query)) $res = '0';
            break;
        default:
            $res = '0 bad "a"';
            break;
    }
    $link->close();
}
else $res = '0 no "a"';

$res = array();//"resp" => $res
$res["resp"] = $res;
header('Content-type: application/json');
echo json_encode( $res , JSON_UNESCAPED_UNICODE);
?>

```

Рис. 20 Реализация добавления новой записи

Кроме того, установка соединения с базой данных была вынесена в отдельную функцию, чтобы при открытии новой страницы можно было вызвать только её. Результат представлен на Рис. 21

```

<?php
$link = mysqli_connect("127.0.0.1", "course", "pass", "vkr");
mysqli_set_charset($link, "utf8");
?>

```

Рис. 21 Реализация соединения с базой данных

В конце было реализовано оформление web-страниц при помощи CSS. Фрагмент представлен на Рис. 22

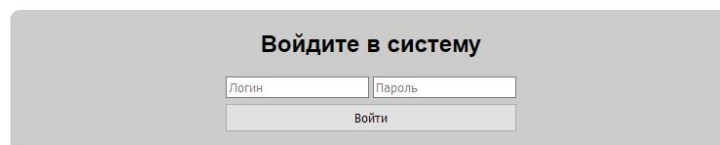
```
.wrap{
  position: absolute;
  width: 700px;
  left: calc(50vw - 350px);
  top: 20px;
  padding: 20px;
  margin-bottom: 20px;
  background-color: rgba(0,0,0,0.2);
  border-radius: 10px;
  font-family: sans-serif;
}
```

Рис. 22 Фрагмент реализации CSS

## 2.2 Описание работы информационной системы по управлению материальной базой предприятия

Разработанная информационная система управления представляет собой web-приложение, в котором сотрудники предприятия могут заносить в базу данных любые объекты материальной базы и быстро получать хранящуюся в базе данных информацию обо всех уже добавленных объектах материальной базы.

Первое окно, которое встречает пользователя, это окно авторизации, представленное на Рис. 23 .



The image shows a login form with a light gray background. At the top, the text "Войдите в систему" is centered. Below it are two input fields: the first is labeled "Логин" and the second is labeled "Пароль". Underneath these fields is a button labeled "Войти".

Рис. 23 Окно авторизации

Затем, если была произведена авторизация пользователем со стандартными правами, будет выведено рабочее окно пользователя, включающее возможность сменить пользователя, выполнить поиск по инвентарному номеру, открыть список всех объектов материальной базы или же добавить новый объект в базу данных, заполнив соответствующую форму. Окно пользователя представлено на Рис. 24

**Управление материальной базой предприятия**

Добро пожаловать, user! ([выход](#))

---

**Поиск по инвентарному номеру**

Введите инвентарный номер

**Объекты**

[Список объектов](#)

Добавить объект:

Введите название объекта

Выберите категорию  | Выберите расположение  | Выберите ответственного

Добавьте описание/характеристики

Инвентарный номер присваивается автоматически!

Рис. 24 Окно пользователя

Если пользователь обладает правами администратора, то его рабочее окно будет несколько отличаться. После действий, доступных обычному пользователю, появится поле с инструментарием, который работает по тем же принципам, что и взаимодействие с материальной базой. Окно администратора представлено на Рис. 25

## Управление материальной базой предприятия

Добро пожаловать, admin! ([выход](#))

---

### Поиск по инвентарному номеру

Проверить

### Объекты

[Список объектов](#)

Добавить объект:

Выберите категорию ▾

Выберите расположение ▾

Выберите ответственного ▾

Инвентарный номер присваивается автоматически!

Добавить

### Возможности администратора

#### Категории

[Список категорий](#)

Добавить категорию:

Добавить

#### Расположения

[Список расположений](#)

Добавить расположение:

Добавить

#### Ответственные

[Список ответственных](#)

Добавить ответственного:

Фамилия

Имя

Отчество

Добавить

#### Пользователи системы

[Список пользователей](#)

Добавить пользователя:

☐ Администратор

Добавить

Рис. 25 Окно администратора

Пользователь может просмотреть список объектов материальной базы, нажав на соответствующую кнопку. Вид данного списка показан на Рис. 26

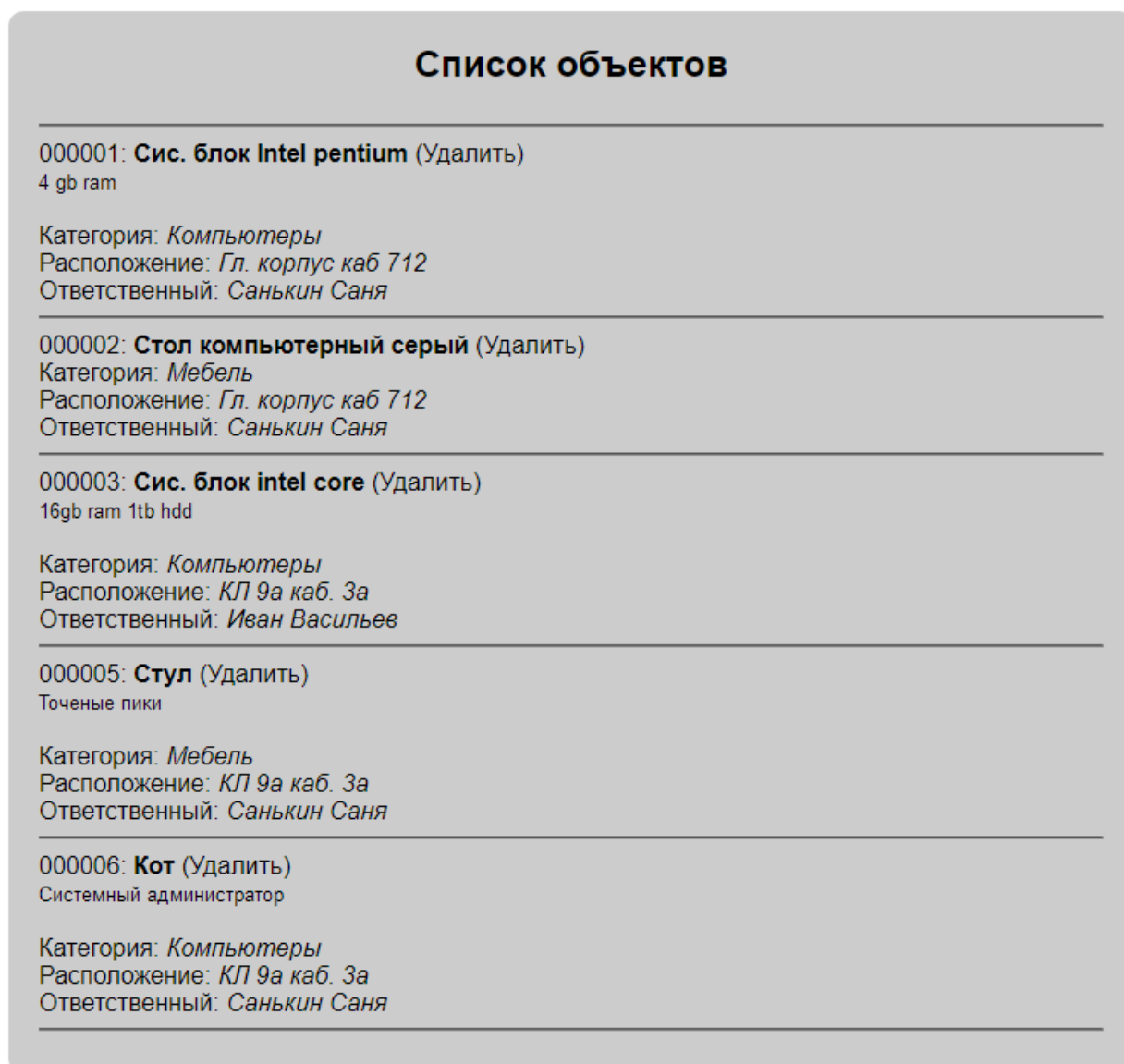


Рис. 26 Окно списка всех объектов

Аналогичным образом организованы списки пользователей, расположений, ответственных людей и категорий объектов, доступные пользователям с правами администратора.

Если же необходимо узнать информацию о конкретном предмете, если известен его инвентарный номер, пользователь может выполнить поиск по инвентарному номеру в соответствующем окне, вид которого представлен на Рис. 27

The screenshot shows a web interface titled "Поиск по инвентарному номеру". It features a search input field containing the number "6" and a "Проверить" button. Below the search bar, the results are displayed: "Объект: Кот инв:000006", "Системный администратор", "Категория: Компьютеры", "Рсположение: КЛ 9а каб. 3а", and "Ответственный: Санькин Саня". At the bottom of the results section is a "Редактировать" button.

Рис. 27 Окно поиска по инвентарному номеру

В результате нами была реализована ИСУ, позволяющая автоматизировать управление материальной базой предприятия, тем самым повысить производительность труда и уменьшить время на обработку информации. Как следствие, образуются дополнительные временные ресурсы на решение новых задач внутри организации.

## 2.3 Руководство пользователя

### 2.3.1 Начало работы.

Для начала работы с системой необходимо иметь в своём распоряжении персональный компьютер с выходом в Internet или локальным соединением с хост-сервером, на котором развернута система, а также браузер с поддержкой HTML5 для работы с web-интерфейсом системы.

После чего необходимо перейти на указанный системным администратором ресурс, для получения доступа к системе.

### 2.3.2 Окно авторизации.

The screenshot shows a login window titled "Войдите в систему". It contains two input fields labeled "Логин" and "Пароль", and a "Войти" button below them.

Рис. 28 Окно входа в систему



Оказавшись на необходимом ресурсе, пользователь увидит окно авторизации (см. Рис. 28 ).

В поле «Логин» необходимо ввести имя учетной записи пользователя, а в поле «Пароль», соответственно, пароль учетной записи.

Затем, необходимо нажать на кнопку «Войти», чтобы подтвердить ввод данных.

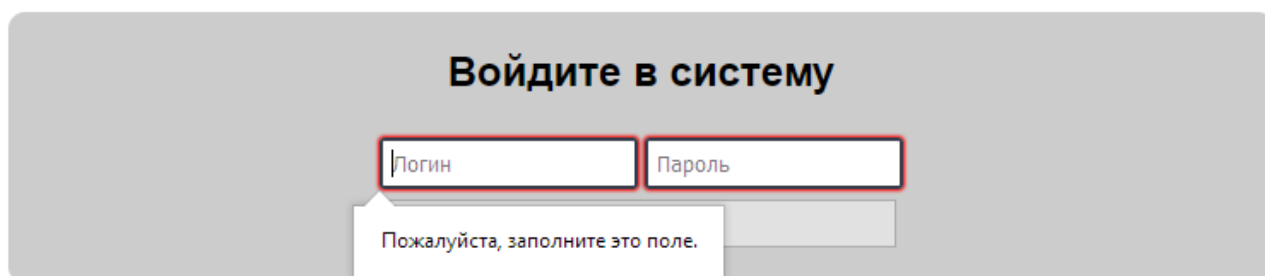


Рис. 29 Окно ошибки: незаполненные поля

В том случае, если пользователь попытается войти без ввода логина или пароля, ему будет выведено диалоговое окно с просьбой заполнить пустые поля (см. Рис. 29 ).

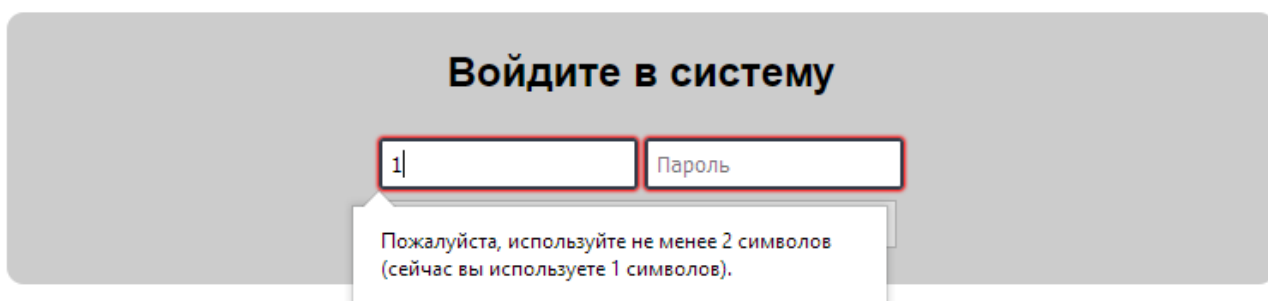


Рис. 30 Окно ошибки: недостаточная длина логина/пароля

В том случае, если пользователь введет слишком короткий логин или пароль, ему будет выведено сообщение о том, что введено слишком малое число символов (см. Рис. 30 ). По-умолчанию, минимальная длина логина и пароля составляет 2 символа.

В том случае, если пользователь введет неверные данные, например, ошибется при вводе логина, он будет возвращен обратно в окно авторизации и получит ошибку «Неверные данные!» (см. Рис. 31 ). В таком случае, пользователю снова предоставляется возможность ввода данных.

**Войдите в систему**

Логин Пароль

Войти

Неверные данные!

Рис. 31 Окно ошибки: неверный ввод данных

### 2.3.3 Рабочее окно пользователя.

Пройдя процедуру авторизации в качестве пользователя без прав администратора, будет выведено рабочее окно пользователя, приведенное на Рис. 32

**Управление материальной базой предприятия**

Добро пожаловать, user! ([выход](#))

---

**Поиск по инвентарному номеру**

Введите инвентарный номер Проверить

**Объекты**

[Список объектов](#)

Добавить объект:

Введите название объекта

Выберите категорию Выберите расположение Выберите ответственного

Добавьте описание/характеристики

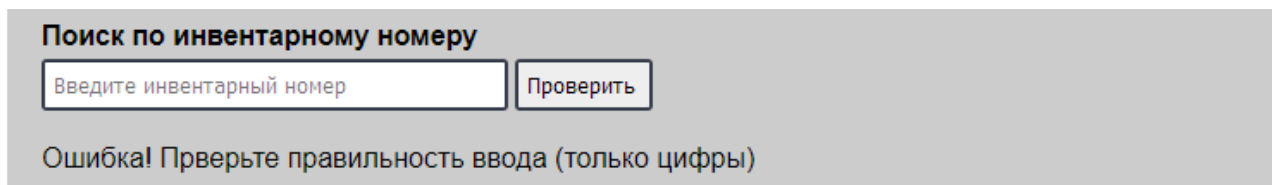
Инвентарный номер присваивается автоматически!

Добавить

Рис. 32 Внешний вид окна пользователя

В самом верху окна пользователь видит текущий логин, под которым был осуществлён вход в систему, а также кнопку «выход», которая вернет его обратно в окно авторизации, где он сможет сменить текущую учетную запись.

Поле «Поиск по инвентарному номеру» позволяет пользователю осуществить поиск в материальной базе предприятия объекта в том случае, если ему известен инвентарный номер требуемого объекта.



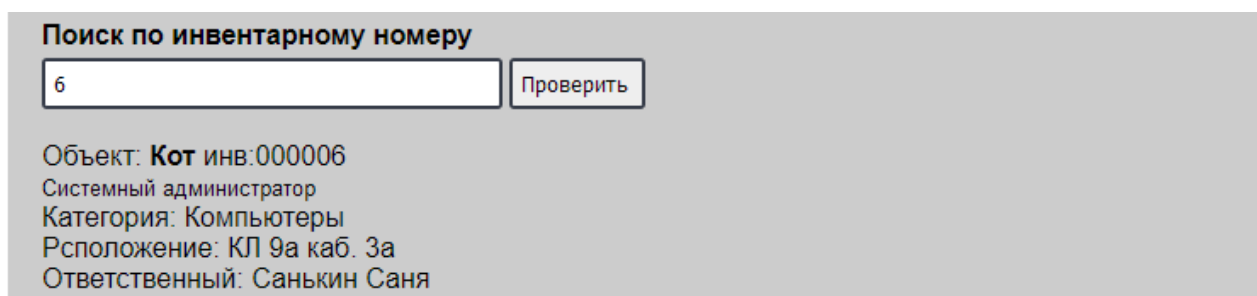
Поиск по инвентарному номеру

Введите инвентарный номер

Ошибка! Прверьте правильность ввода (только цифры)

Рис. 33 Вывод ошибки о неправильности ввода

Если пользователь не ввёл никаких данных или же ввёл их неверно, будет выведено сообщение об ошибке (см. Рис. 33 ). В таком случае, пользователь может снова ввести инвентарный номер.



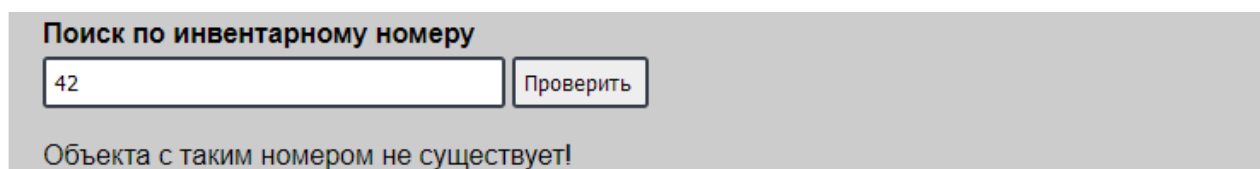
Поиск по инвентарному номеру

6

Объект: **Кот** инв:000006  
Системный администратор  
Категория: Компьютеры  
Рсположение: КЛ 9а каб. 3а  
Ответственный: Санькин Саня

Рис. 34 Вывод информации об объекте

В том случае, если данные были введены корректно и объект с таким номером действительно существует, пользователь получит информацию о нём, которая будет выведена под полем ввода инвентарного номера (см. Рис. 34 ).



Поиск по инвентарному номеру

42

Объекта с таким номером не существует!

Рис. 35 Вывод ошибки об отсутствии объекта

Если данные были введены корректно, но объекта с таким номером не существует в базе данных, пользователь будет извещен об этом (см. Рис. 35 ).

Следующий раздел позволяет просмотреть все объекты, которые в данный момент находятся в материальной базе, по нажатию кнопки «Список объектов». Внешний вид выводимого списка представлен на Рис. 36 .

## Список объектов

000001: **Сис. блок Intel pentium** (Удалить)

4 gb ram

Категория: *Компьютеры*

Расположение: *Гл. корпус каб 712*

Ответственный: *Санькин Саня*

000002: **Стол компьютерный серый** (Удалить)

Категория: *Мебель*

Расположение: *Гл. корпус каб 712*

Ответственный: *Санькин Саня*

000003: **Сис. блок intel core** (Удалить)

16gb ram 1tb hdd

Категория: *Компьютеры*

Расположение: *КЛ 9а каб. 3а*

Ответственный: *Иван Васильев*

000005: **Стул** (Удалить)

Точеные пики

Категория: *Мебель*

Расположение: *КЛ 9а каб. 3а*

Ответственный: *Санькин Саня*

000006: **Кот** (Удалить)

Системный администратор

Категория: *Компьютеры*

Расположение: *КЛ 9а каб. 3а*

Ответственный: *Санькин Саня*

000007: **Монитор** (Удалить)

Категория: *Компьютеры*

Расположение: *КЛ9, ауд. 8*

Ответственный: *Санькин Саня*

Рис. 36 Список объектов

Каждый объект в данном списке представлен в следующем формате:

- Номер: Название
- Описание
- Категория
- Расположение
- Ответственный персонал

Кроме того, у пользователя существует возможность удалить объект из базы нажатием кнопки «Удалить» рядом с названием нужного объекта. После этого пользователю будет выведено окно с запросом подтверждения действия (см. Рис. 37 ).

При подтверждении, объект будет удалён, а текущая страница обновится для отображения обновлённого списка.

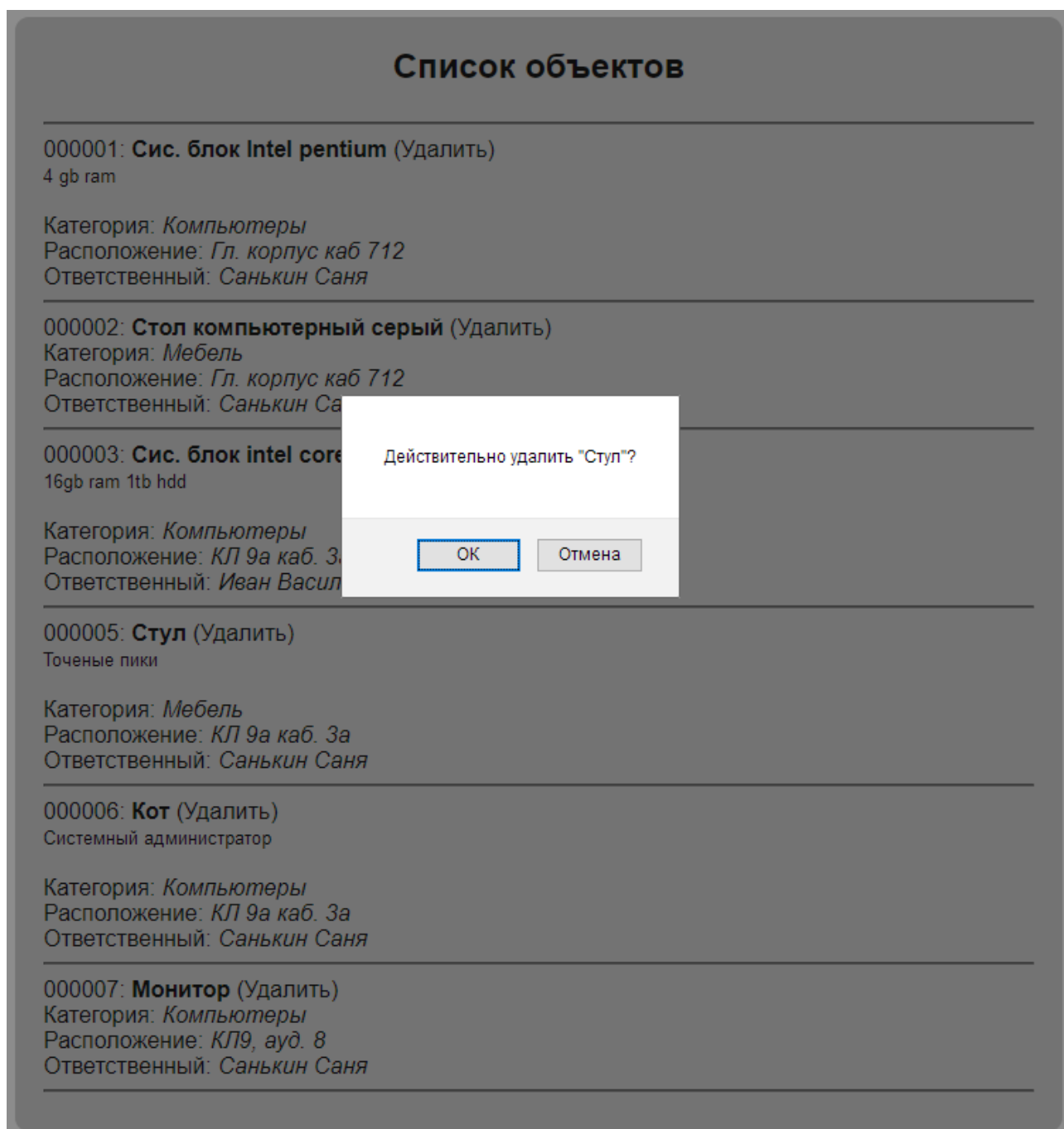


Рис. 37 Запрос подтверждения удаления

Для возвращения обратно к окну пользователя, необходимо нажать на стрелку в левом углу экрана (см. Рис. 38 ). Подобное правило применимо ко всем остальным окнам, открываемым вместо окна пользователя, поэтому в дальнейшем не будут рассматриваться.



Рис. 38 Обозначение возврата на страницу назад

Последней секцией рабочего окна пользователя является добавление объекта в материальную базу (см. Рис. 39 ).

Добавить объект:

Введите название объекта

Выберите категорию | Выберите расположение | Выберите ответственного

Добавьте описание/характеристики

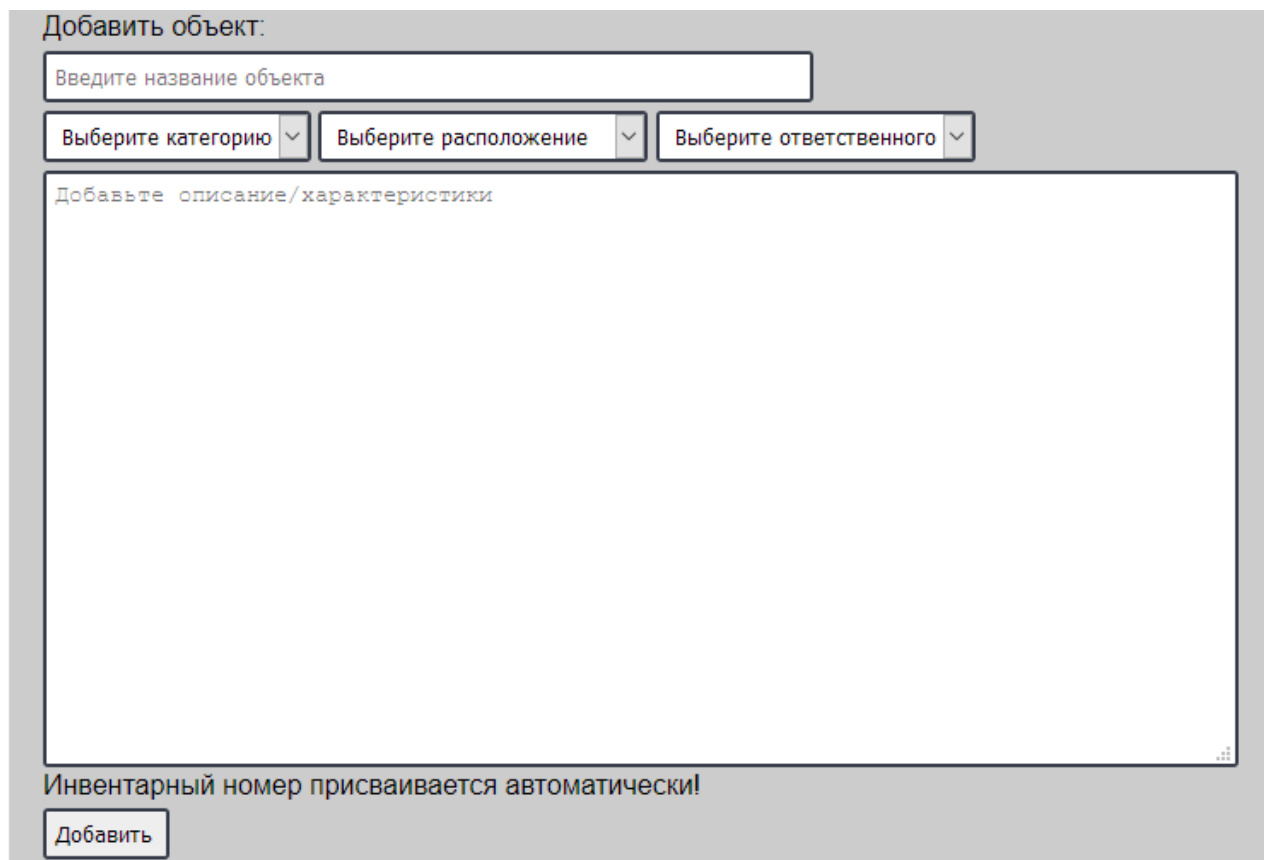
Инвентарный номер присваивается автоматически!

Добавить

Рис. 39 Раздел добавления объекта

Пользователю необходимо ввести название нового объекта, выбрать из выпадающего списка категорию, к которой объект относится, его расположение и лицо, ответственное за данный объект, а затем, если в этом есть необходимость, добавить короткое описание объекта, содержащее какие-либо дополнительные характеристики объекта. Кроме того, поле ввода описания возможно изменить в размере для более удобного ввода каких-либо данных, однако, предполагается, что описание будет достаточно коротким и

информативным. Пример изменения размера окна ввода приведен на Рис. 40 .  
Пример ввода данных приведен на Рис. 41 .



Добавить объект:

Введите название объекта

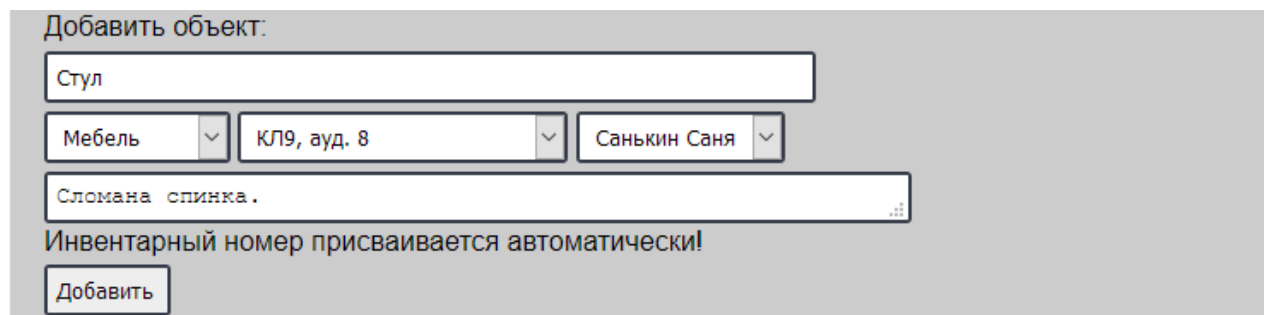
Выберите категорию ▾    Выберите расположение ▾    Выберите ответственного ▾

Добавьте описание/характеристики

Инвентарный номер присваивается автоматически!

Добавить

Рис. 40 Изменение размера окна ввода описания



Добавить объект:

Стул

Мебель ▾    Кл9, ауд. 8 ▾    Санькин Саня ▾

Сломана спинка.

Инвентарный номер присваивается автоматически!

Добавить

Рис. 41 Пример заполнения полей ввода

После ввода всех необходимых данных, пользователю необходимо нажать на кнопку «Добавить». После чего справа от кнопки «Добавить» появится сообщение об успешности добавления объекта или же об ошибке.

### 2.3.4 Рабочее окно администратора.

## Управление материальной базой предприятия

Добро пожаловать, admin! ([выход](#))

### Поиск по инвентарному номеру

### Объекты

[Список объектов](#)

Добавить объект:

Инвентарный номер присваивается автоматически!

## Возможности администратора

### Категории

[Список категорий](#)

Добавить категорию:

### Расположения

[Список расположений](#)

Добавить расположение:

### Ответственные

[Список ответственных](#)

Добавить ответственного:

### Пользователи системы

[Список пользователей](#)

Добавить пользователя:

☐ Администратор



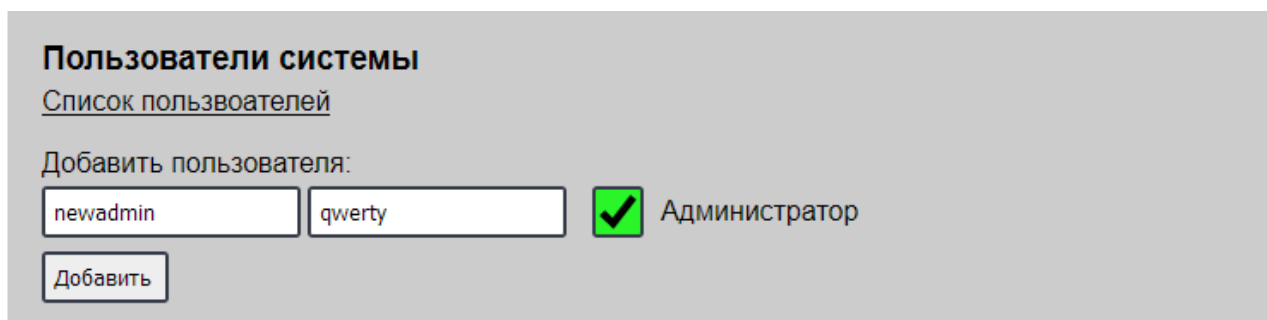
Рис. 42 Рабочее окно администратора

Если пользователь, вошедший в систему, имеет права администратора, то он будет перемещён на страницу рабочего окна администратора (см. Рис. 42 ).

Администратор может выполнять те же функции, что и пользователь, функции которого рассмотрены в соответствующем разделе (см. Рабочее окно пользователя.), однако, имеет дополнительный функционал, размещенный в разделе «Возможности администратора».

Методика работы дополнительных полей схожа с полем «Добавление объекта» в рабочем окне пользователя. Администратор может просмотреть список интересующего его блока данных и добавить новую запись.

Рассмотрим на примере добавления нового пользователя с правами администратора (см. Рис. 43 ) и просмотра списка пользователей (см. Рис. 44 ).



**Пользователи системы**  
Список пользователей

Добавить пользователя:


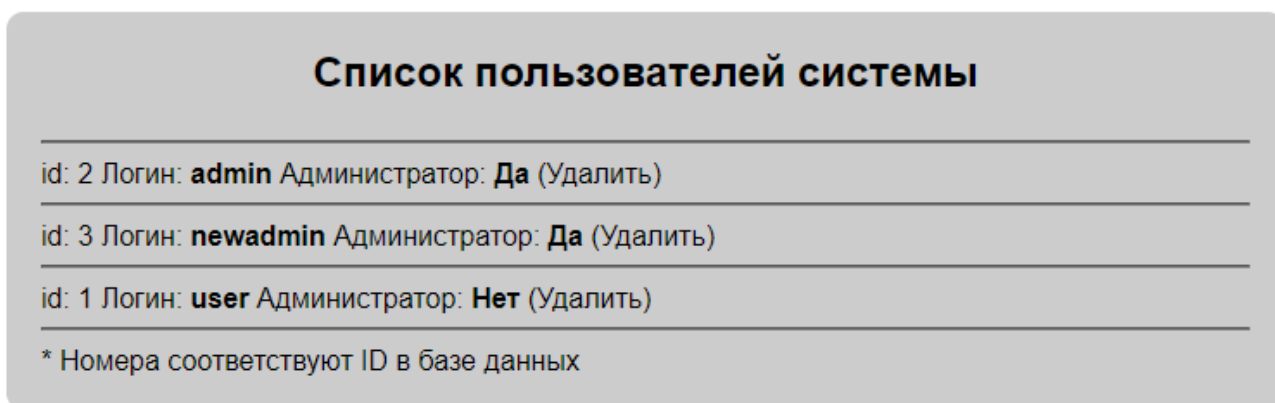
 Администратор

Рис. 43 Пример добавления нового пользователя с правами администратора



**Список пользователей системы**

id: 2	Логин: <b>admin</b>	Администратор: <b>Да</b>	(Удалить)
id: 3	Логин: <b>newadmin</b>	Администратор: <b>Да</b>	(Удалить)
id: 1	Логин: <b>user</b>	Администратор: <b>Нет</b>	(Удалить)

\* Номера соответствуют ID в базе данных

Рис. 44 Пример списка пользователей

## **Заключение**

В процессе выполнения работы в рамках сформулированных задач было проделано следующее:

1. Проведен анализ состояния проблемы и подходов к ее решению. В ходе работы было определено, что необходимо оптимизировать деятельность организации путем создания информационной системы управления, позволяющей обеспечить управление материальной базой предприятия.
2. Был произведен и обоснован выбор технологии реализации и необходимых программных платформ.
3. Была реализована информационная система управления при помощи web-сервера OpenServer, СУБД MySQL и языка программирования PHP, полностью соответствующая представленному техническому заданию.
4. В соответствии с поставленными задачами подготовлена техническая и сопроводительная документация, описывающая порядок работы с системой.

## Список информационных источников

1. HTML // Википедия URL: <https://ru.wikipedia.org/wiki/HTML> (дата обращения: 14.05.2020).
2. MySQL Workbench // Википедия URL: [https://en.wikipedia.org/wiki/MySQL\\_Workbench](https://en.wikipedia.org/wiki/MySQL_Workbench) (дата обращения: 31.05.2020).
3. UML // Википедия URL: <https://ru.wikipedia.org/wiki/UML> (дата обращения: 31.05.2020).
4. phpMyAdmin // Википедия URL: <https://ru.wikipedia.org/wiki/PhpMyAdmin> (дата обращения: 31.05.2020).
5. Белов Ю.С., Бурмистров А.В. Недостатки реляционных баз данных // Электронный журнал: наука, техника и образование. 2015. №3 (3). С. 29-38. URL: <http://nto-journal.ru/uploads/articles/aaf4116dd0b56a4e2e6df6a597c753bf.pdf> (дата обращения 25.04.2020)
6. Бэнкер, К. MongoDB в действии / К. Бэнкер. –М.: Высшаяшкола, 2016. -287с
7. ГОСТ 19.502-78. Описание применения. Требования к содержанию и оформлению; Введ. 1980-01-01. М.: Стандартинформ. 2 с. (Единая система программной документации).
8. ГОСТ 19.503-79. Руководство системного программиста. Требования к содержанию и оформлению; Введ. 1980-01-01. М.: Стандартинформ. 3 с. (Единая система программной документации).
9. ГОСТ 19.504-79. Руководство Программиста. Требования к содержанию и оформлению; Введ. 1980-01-01. М.: Стандартинформ. 3 с. (Единая система программной документации).
10. ГОСТ 19.505-79. Руководство Оператора. Требования к содержанию и оформлению; Введ. 1980-01-01. М.: Стандартинформ. 3 с. (Единая система программной документации).

11. ГОСТ 2.105-95. Общие требования к текстовым документам. Взамен ГОСТ 2.105-79, ГОСТ 2.906-71; Введ. 1996-07-01. М.: Всероссийский научно-исследовательский институт стандартизации и сертификации в машиностроении (ВНИИНМАШ). 27 с. (Межгос. стандарт. Единая система конструкторской документации).
12. ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание. Взамен ГОСТ 7.0-84; Введ. 2004-07-01. Межгосударственный совет по стандартизации, метрологии и сертификации. М. : Изд-во стандартов. 111 с. (Межгос. стандарт. Система стандартов по информации, библиотечному и издательскому делу. Общие требования и правила составления).
13. ГОСТ Р 52653-2006. Информационно-коммуникационные технологии в образовании. Термины и определения; Введ. 2008-07-01. М.: Стандартинформ. 18 с. (Национальный Стандарт Российской Федерации).
14. ГОСТ Р 52657-2006. Образовательные интернет-порталы федерального уровня. Рубрикация информационных ресурсов; Введ. 2008-07-01. М.: Стандартинформ. 10 с. (Информационно-коммуникационные технологии в образовании).
15. ГОСТ Р 53620-2009. Электронные образовательные ресурсы. Информационно-коммуникационные технологии в образовании; Введ. 2011-01-01. М.: Стандартинформ. 10 с. (Информационно-коммуникационные технологии в образовании).
16. ГОСТ Р 53625-2009. Обучение, образование и подготовка, менеджмент качества, обеспечение качества и метрики. Общий подход; Введ. 2011-01-01. М.: Стандартинформ. 28 с. (Информационная технология).
17. ГОСТ Р 7.0.5-2008 СИБИД. Библиографическая Ссылка. Общие требования и правила составления; Введ. 2009-01-01. М.: Стандартинформ. 35 с. (Система стандартов по информации, библиотечному и издательскому делу).

18. ГОСТ Р 7.0.83-2013. Электронные издания. Основные виды и выходные сведения. С изм. и допол. в ред. от 12.09.2018.; Введ. 2013-10-15. М. : Стандартинформ. 16 с. (Система стандартов по информации, библиотечному и издательскому делу).
19. ГОСТ Р ИСО 9241-3-2003. Эргономические требования при выполнении офисных работ с использованием видеодисплейных терминалов (ВДТ). Требования к визуальному отображению информации; Введ. 2004-01-01. М.: ИПК Издательство стандартов. 62 с. (Государственный стандарт Российской Федерации).
20. ГОСТ Р ИСО 9241-8-2007. Эргономические требования при выполнении офисных работ с использованием видеодисплейных терминалов (Вдт). Требования к отображаемым цветам; Введ. 2009-01-01. М.: Стандартинформ. 45 с. (Национальный стандарт российской федерации).
21. Документация к PostgreSQL 9.6.18 // PostgresPro URL:  
<https://postgrespro.ru/media/docs/postgresql/9.6/ru/postgres-A4-fop.pdf> (дата обращения: 25.04.2020).
22. Драч В.Е., Родионов А.В., Чухраева А.И. Выбор системы управления базами данных для информационной системы промышленного предприятия // Электромагнитные волны и электронные системы. 2018. Т. 23. № 3. С. 71-80.
23. Информационные системы управления // Справочник URL:  
[https://spravochnik.ru/informacionnye\\_tehnologii/setevye\\_informacionnye\\_sistemy/informacionnye\\_sistemy\\_upravleniya/](https://spravochnik.ru/informacionnye_tehnologii/setevye_informacionnye_sistemy/informacionnye_sistemy_upravleniya/) (дата обращения: 18.04.2020).
24. Л. Ю. Сапожникова Логистика: конспект лекций. - Новокузнецк: НФИ КемГУ, 2013. - 92 с.
25. Оформитель библиографических ссылок // SNOSKA.INFO URL:  
<http://snoskainfo.ru/> (дата обращения: 10.02.2020).

26. Разработка Web-приложений на PHP и MySQL: Пер. с англ./Л. Томсон, Л. Веллинг. –К.: Издательство «ДиаСофт», 2010. —672 с.
27. СанПиН 2.4.2.2821-10. Санитарно-эпидемиологические правила и нормативы. С изм. и допол. в ред. от 29 июня 2011 г., 25 декабря 2013 г., 24 ноября 2015 г.; Введ. 2011-09-01. М.: Роспотребнадзор. 72 с. (Санитарно-эпидемиологические требования к условиям и организации обучения в общеобразовательных организациях).
28. Федеральный закон от 27.07.2006 N 149-ФЗ (ред. от 03.04.2020) "Об информации, информационных технологиях и о защите информации" // КонсультантПлюс URL:  
[http://www.consultant.ru/document/cons\\_doc\\_LAW\\_61798/c5051782233acca771e9adb35b47d3fb82c9ff1c/](http://www.consultant.ru/document/cons_doc_LAW_61798/c5051782233acca771e9adb35b47d3fb82c9ff1c/) (дата обращения: 14.04.2020).